SWKM 2008: Social Web and Knowledge Management, Proceedings

CEUR Workshop Proceedings

Dolog, Peter; Kroetzsch, Markus; Schaffert, Sebastian; Vrandecic, Denny

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

Proceedings of the 2008 Workshop on

Social Web and Knowledge Management

— SWKM 2008 —

http://km.aifb.uni-karlsruhe.de/ws/swkm2008

Located at the

17th World Wide Web Conference

WWW 2008

April 22nd, 2008

Beijing, China

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

Preface

Welcome to the 2008 Workshop on Social Web and Knowledge Management (SWKM 2008). The focus of this event is the *social web*, possibly the most interesting part of the Web 2.0, which aims at facilitating and enriching means of interaction among people on the Web. It is characterised by a strong emphasis of communities where people share experiences, information and knowledge, meet and discuss, or do business together. *Knowledge management* systems, with their core focus on knowledge exchange and experience sharing, can benefit from the advancement of the social web. Both areas share a common interest in social structures and social computing, and web-centred approaches can provide the underlying platform for innovative knowledge management systems.

SWKM 2008 brings together people from the areas of social web technologies, semantic systems, and knowledge management. The workshop's major goal is to study and elaborate possible synergies between social computing, social web, semantic technologies, and knowledge management, and to provide a glimpse at the state of the art in the area.

The SWKM 2008 workshop has received fifteen submissions out of which six were selected for presentation and publication in these proceedings. The accepted contributions span across various topics such as generation of user profiles from folksonomies, maintenance costs for large hyperstructures in wikis, computation of access permissions based on social networks, tagging, link sharing, and service integration for cultural heritage.

We are grateful for the dedicated work of both authors and reviewers who contributed their time to ensure the good quality of the technical program. The organisation of this event was made possible through the support of the European Union in the research projects *Active* (`http://active-project.eu`) and *KiWi* (`http://kiwi-project.eu`).

April 2008

Peter Dolog
Markus Krötzsch
Sebastian Schaffert
Denny Vrandečić

SWKM 2008

http://km.aifb.uni-karlsruhe.de/ws/swkm2008

Programme Committee

- Harith Alani, University of Southampton, UK
- Anupriya Ankolekar, HP Labs, Palo Alto, USA
- Francois Bry, LMU Munich, Germany
- John Davies, BT, Ipswich, UK
- Norbert Eisinger, LMU Munich, Germany
- Hans-Jörg Happel, FZI, Karlsruhe, Germany
- Tom Heath, Open University, UK
- Martin Hepp, STI Innsbruck, University of Innsbruck, Austria
- Nick Kings, BT, Ipswich, UK
- Hong-Gee Kim, Seoul National University, South Korea
- Qing Li, City University of Hong Kong, China
- Peter Mika, Yahoo! Research, Barcelona, Spain
- Peter Axel Nielsen, AAU Aalborg, Denmark
- Natasha Noy, Stanford University, USA
- Eyal Oren, VU Amsterdam, The Netherlands
- Valentina Presutti, Institute of Cognitive Sciences and Technology (CNR), Italy
- Timothy K. Shih, Tamkang University, Taiwan
- Elena Simperl, STI Innsbruck, University of Innsbruck, Austria
- Pavel Smrz, BUT Brno, Czech Republic
- York Sure, SAP Research, CAC Karlsruhe, Germany
- Hideaki Takeda, Tokyo Research Institute, Japan
- Marcel Tilly, European Microsoft Innovation Center, Aachen, Germany
- Tania Tudorache, Stanford University, USA

Additional Reviewer

- Alex Kohn, LMU Munich, Germany

Organisation Commitee

- Peter Dolog, Computer Science Department, Aalborg University, Denmark
- Markus Krötzsch, Institut AIFB, Universität Karlsruhe (TH), Germany
- Sebastian Schaffert, Salzburg Research, Austria
- Denny Vrandečić, Institut AIFB, Universität Karlsruhe (TH), Germany

Contents

A Study of User Profile Generation from Folksonomies

Ching-man Au Yeung
cmay06r@ecs.soton.ac.uk

Nicholas Gibbins
nmg@ecs.soton.ac.uk

Nigel Shadbolt
nrs@ecs.soton.ac.uk

Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, United Kingdom

ABSTRACT

Recommendation systems which aim at providing relevant information to users are becoming more and more important and desirable due to the enormous amount of information available on the Web. Crucial to the performance of a recommendation system is the accuracy of the user profiles used to represent the interests of the users. In recent years, popular collaborative tagging systems such as del.icio.us have aggregated an abundant amount of user-contributed metadata which provides valuable information about the interests of the users. In this paper, we present our analysis on the personal data in folksonomies, and investigate how accurate user profiles can be generated from this data. We reveal that the majority of users possess multiple interests, and propose an algorithm to generate user profiles which can accurately represent these multiple interests. We also discuss how these user profiles can be used for recommending Web pages and organising personal data.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software; H.3.5 [**Information Storage and Retrieval**]: Online Information Services; H.5 [**Information Interfaces and Presentation (I.7)**]:

General Terms

Algorithms, Experimentation, Human Factors

Keywords

collaborative tagging, folksonomy, personomy, user profile

1. INTRODUCTION

The amount of resources on the Web nowadays is so enormous that retrieval of relevant information is getting more and more difficult. While users are desperate to obtain information that is relevant to their needs and to avoid information that are irrelevant, publishers of resources are also eager to deliver their information to their targeted readers. This has resulted in the rise of recommendation systems [3] which aim to recommend relevant and interesting resources to users. An important aspect of user profiles is whether they can truly reflect the interests or expertise of the users. Some research works attempt to construct user profiles based on the browsing history of the users [9, 22], or on the documents collected by the users [4].

Recently, the rising popularity of collaborative tagging systems [8], such as del.icio.us[1] and Flickr[2], has provided new sources for understanding the interests of Web users. Collaborative tagging systems allow users to choose their own words as tags to describe their favourite Web resources, resulting in an emerging classification scheme now commonly known as a *folksonomy* [24]. Given that the resources and the tags posted by Web users to these systems are supposed to be highly dependent on their interests, folksonomies thus provide rich information for building more accurate and more specific user profiles for use in various applications.

Currently, only a few studies in the literature try to construct user profiles from data in collaborative tagging systems [5, 13], and usually only a single set of popular tags are used to represent user interests. However, we observe that tags used by users are very diverse and span across many different domains. This implies that users usually have a wide range of interests. Therefore, a single set of tags may not be the most suitable representation of a user profile, as it is not able to reflect the multiple interests of users. In this paper, we propose a network analysis technique performed on the personomy [11] of a user to identify the different interests of a user, and to construct a more comprehensive user profile based on the results. Evaluations show that our algorithm is able to reveal the different domains in which the users are interested, and construct more informative and specific user profiles.

This paper is structured as follows. Section 2 introduces folksonomies and personomies. Section 3, presents the analysis of the data collected from del.icio.us which motivated this research. In Section 4, we describe in detail our proposed algorithm for user profile construction. Evaluations, discussions and potential applications are presented in Section 5. We mentioned related works in Section 6. Finally, Section 7 concludes the paper and gives future research directions.

2. FOLKSONOMIES AND PERSONOMIES

Folksonomies [24] are user-contributed data aggregated by collaborative tagging systems. In these systems, users are allowed to choose terms freely to describe their favourite Web resources. A folksonomy is generally considered to consist

[1]http://del.icio.us/
[2]http://www.flickr.com/

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.

of at least three sets of elements, namely users, tags and resources. Although there can be different kinds of resources, in this article we will focus on Web documents, such as those being bookmarked in del.icio.us. Formally, a folksonomy is defined as follows [15].

Definition 1. A folksonomy $\mathbf{F}$ is a tuple $\mathbf{F} = (U, T, D, A)$, where U is a set of users, T is a set of tags, D is a set of Web documents, and $A \subseteq U \times T \times D$ is a set of annotations.

If we want to understand the interests of a single user, we only need to concentrate on the tags and documents that are associated with this particular user. Such set of data is given the name ***personomy*** [11].[3]

Definition 2. A personomy $\mathbf{P}_u$ of a user u is a restriction of a folksonomy $\mathbf{F}$ to u: i.e. $\mathbf{P}_u = (T_u, D_u, A_u)$, where A_u is the set of annotations of the user: $A_u = \{(t,d)|(u,t,d) \in A\}$, T_u is the user's set of tags: $T_u = \{t|(t,d) \in A_u\}$, and D_u is the user's set of documents: $D_u = \{d|(t,d) \in A_u\}$.

This definition is identical to the one mentioned in [11], except that we choose to exclude the sub-tag/super-tag relation, since most collaborative tagging systems do not offer such functionality and we will not deal with this here.

To perform analysis on the personomy of a user, we first represent the personomy in the form of a network, with nodes representing tags and documents associated with the user. If folksonomy can be considered as a hypergraph with three disjoint sets of nodes (user, tags and documents), a personomy can be represented as a bipartite graph by extracting the part that is related to the user. The bipartite graph TD_u of a personomy of a user u is defined as follows.

$$TD_u = \langle T_u \cup D_u, E_{td} \rangle, E_{td} = \{(t,d)|(t,d) \in A_u\}$$

An edge exists between a tag and a document if the tag is assigned to the document. The graph can be represented in matrix form, which we denote as $\mathbf{X} = \{x_{ij}\}$, $x_{ij} = 1$ if there is an edge connecting t_i and d_j, and $x_{ij} = 0$ otherwise.

To perform document clustering, we can fold the bipartite graph into a one-mode network [15] of documents: $\mathbf{D} = \mathbf{X}'\mathbf{X}$. The adjacency matrix $\mathbf{D}$ represents the personal repository of the user. Links between documents are weighted by the number of tags that have been assigned to both documents. Thus, documents with higher weights on the links between them can be considered as more related. On the other hand, a one-mode network of tags can be constructed in a similar fashion: $\mathbf{T} = \mathbf{X}'\mathbf{X}$. $\mathbf{T}$ represents semantic network which shows the associations between different tags. In other words, this is the personal vocabulary or a simple ontology used by the particular user.

To facilitate the following discussions, we further define several notations here. Firstly, we denote the set of documents tagged by the tag t in the personomy of user u by $D_{u,t}$:

$$D_{u,t} = \{d|(t,d) \in A_u\}$$

Also, we define $Co_u(t_1, t_2)$ which indicates whether two tags t_1 and t_2 have been used on the same document by a user:

$$Co_u(t_1,t_2) = \begin{cases} 1 & \text{if } (t_1,d) \in A_u,\ (t_2,d) \in A_u \text{ for some } d \\ 0 & \text{otherwise} \end{cases}$$

[3]In the blogosphere, the term personomy has also been used in a more general sense to represent the aggregated digit manifestation of a user on the Web. See http://personomies.com/what-are-personomies/.

Total number of users		9,185
Tags	Maximum	18,952
	Minimum	1
	Mean	285
Bookmarks	Maximum	34,201
	Minimum	1
	Mean	602

Table 1: Summary of data obtained from del.icio.us.

3. ANALYSIS OF PERSONOMIES

To understand the characteristics of personomies in collaborative tagging systems, we perform analysis on data collected from del.icio.us. In particular, we want to gain insight into the general behaviour of Web users using these systems. We also want to understand if users are generally interested in a rather specific domain, such as we might expect when studying the publications of a researcher, or if they are more likely to be interested in a wide range of topics.

In December 2007, we collected the bookmarking data of 9,431 users of del.icio.us, including their bookmarks and the tags they used, by crawling del.icio.us user names which appeared on the page showing the recently updated bookmarks.[4] It is noted that among the 9,431 users whose data we have collected, 246 of them apply no tags to any of their stored bookmarks. These users are filtered when performing the following analysis. We summarise the statistics of the data of the remaining 9,185 users in Table 1 and Figure 1.

3.1 Number of Tags and Bookmarks of a User

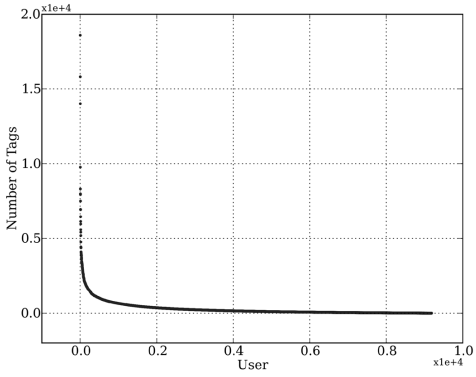
From the summary of the data in Table 1 and Figure 1, we can see that on average a user have used 285 unique tags and have saved 602 unique bookmarks on del.icio.us. Although some users have over 18,000 tags and over 34,000 bookmarks, only a very small number of users have more than a thousand tags or bookmarks. This finding agrees with what Golder and Huberman [8] report in their paper, showing that there are a small number of users having a large number of tags and bookmarks, and a large number of users having a small number of tags and bookmarks, suggesting a power-law distribution.

In addition, we examine the correlation between the number of tags and the number of bookmarks of the users. Figure 2 shows a scatter plot of the data. It shows a moderate relationship between the number of tags and the number of bookmarks, with a correlation coefficient of 0.55.

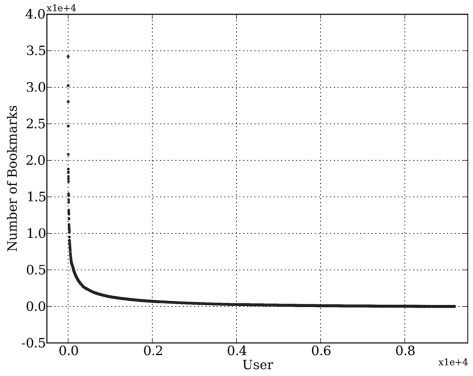
In fact, it is natural to suggest that when there are more bookmarks more tags are required to distinguish between different bookmarks by putting them into more specific categories. However the bookmarks and tags of the users in the system are also highly dependent on the interests of the users. If a user has a very specific interest, a small number of tags will be enough for even a large number of bookmarks, as they will probably be about the same topic. On the other hand, if a user has diverse interests, more tags may be required to describe even a small number of bookmarks.

A further investigation of the data reveals that the correlation between the two numbers is stronger for users with fewer bookmarks than those with many bookmarks. For users with fewer than 500 bookmarks, the correlation coefficient is 0.43. For users with more than 5,000 bookmarks, the

[4]http://del.icio.us/recent



(a) Tags



(b) Bookmarks

Figure 1: Number of tags and bookmarks of the users.

correlation coefficient is only 0.14. A similar result can also be found in [8]. This may suggest that users with many bookmarks can behave very differently: while some may stick to using a small number of tags on new bookmarks, others may continue to introduce new tags.

3.2 Multiple Interests of Users

With the average number of bookmarks significantly larger than the average number of tags being used, it is obvious that users are very likely to use a tag to describe more than one bookmark. However, the usage of tags also depends on the diversity of interests of the users. A user with only one or two specific interests is likely to use fewer tags than another user who is interested in topics across several different domains. To understand this aspect of users in collaborative tagging system, we propose two measures which reflect the diversity of interests of the users. We will give examples based on the two fictional users in Table 2, one with rather specific interests in Semantic Web related topics, while another has more diverse interests such as cooking and sports.

Firstly, we study the relations between the tags and the bookmarks. If the tags used by a user are all assigned to most of the bookmarks, the user is likely to have a rather specific interest, because this set of tags applies to most of

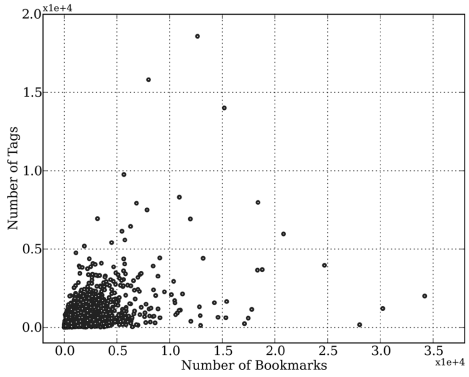


Figure 2: Scatter plot of number of tags against number of bookmarks.

user	bookmark	tags
u_1	d_1	web2.0, semanticweb, ontology, notes
	d_2	semanticweb, ontology
	d_3	semanticweb, ontology, RDF
u_2	d_4	semanticweb, folksonomy, tagging
	d_5	toread, cooking, recipe, food
	d_6	sports, football, news

Table 2: Two example users with their personomies.

the documents that the user is interested in. On the other hand, if most of the tags are only used on a small fraction of bookmarks, it is likely that the user has a broader range of interests. To quantify this characteristic, we propose a measure called ***tag utilisation*** which is defined as follows.

Definition 3. Tag utilisation (TU) of a user u is the average of the fractions of bookmarks on which a tag is used:

$$TagUtil(u) = \frac{1}{|T_u|} \sum_{t \in T_u} \frac{|D_{u,t}|}{|D_u|} \quad (1)$$

In addition, the diversity of a user's interest can also be understood by examining tag co-occurrence. If for a user the tags are always used together with each other, it is likely that the tags are about similar topics, and so the user should have a rather specific interest. If on the other hand the tags are mostly used separately, they are more likely to be about different topics, and thus reflect that the user should have multiple interests which are quite distinctive from each other. Such characteristic can be measure by ***average tag co-occurrence ratio***, which is defined as follows.

Definition 4. Average tag co-occurrence ratio (ATCR) of a user measures how likely two tags are used together on the same bookmark by a user:

$$Avg\_Tag\_Co(u) = \sum_{t_i, t_j \in T_u, t_i \neq t_j} \frac{Co(t_i, t_j)}{2 \times C_2^{|T_u|}} \quad (2)$$

If we represent the co-occurrences between the tags as a network (by constructing the adjacency matrix $\mathbf{T}$), we can

	MAX	MIN	MEAN	STD
TU	1.0000	0.0003	0.0617	0.1388
ATCR	1.0000	0.0000	0.0707	0.1297

Table 3: Summary of the two measures of the data.

see that the average tag co-occurrence ratio is actually equivalent to the density of the network of tags: $Co(t_i, t_j)$ counts the number of edges in the network, while $C_2^{|T_u|}$ calculates the number of possible edges based on the number of nodes. This agrees with the formula of the density of a network:

$$Density = \frac{2 \times |E|}{|V| \times (|V| - 1)} \quad (3)$$

where E is the set of edges and V is the set of nodes. Hence, the average tag co-occurrence ratio actually reflects the cohesion [25] of the network of tags, which in turn reflects whether the tags are related to a specific domain or a wide range of topics.

As an illustrating example, we apply these two measures to the two users listed in Table 2. The tag utilisation of u_1 is 0.60, while that of u_2 is 0.33. The average tag co-occurrence ratio of u_1 is 0.80, while that of u_2 is 0.27. For both measures, u_1 scores higher than u_2, this agrees with the fact that the interests of u_2 are more diverse as observed from this user's bookmark collection.

Next, we apply these two measures on the set of data that we have collected from del.icio.us. The results are summarised in Table 3 and Figure 3.

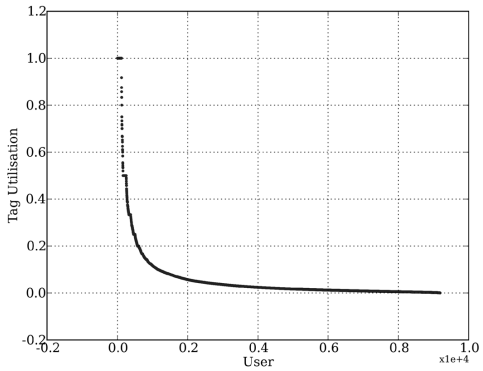
Although the two measures are designed to measure different characteristics of personomies, the results do have very common features. Firstly, the mean values of tag utilisation and average tag co-occurrence ratio both very low, at 0.06 and 0.07 respectively, even though the values span across the whole range from 0 to 1. These values mean that on average a tag is only used on 6% of the bookmarks in a user's collection, and that a tag is only used together with 7% of other tags. We can see that there is a small group of points in both graphs in Figure 3 which attain a value of 1. These actually correspond to users who have only one bookmark in their collection. Other than these the values drop quickly, and the majority of personomies have values less than 0.2 (93% in both measures). Also, there is a strong correlation between tag utilisation and average tag co-occurrence ratio, with a correlation coefficient of 0.71.

Given these figures, we reveal that for most users many tags are used only on a small portion of their bookmarks, and that these tags are not always used together. This suggests that the bookmarks of the users have topics which are rather diverse such that tags do not apply to all of them. Also, a user's tags can be terms from different domains which are not used together very often on the bookmarks. Hence, this indicates that users of del.icio.us have diverse interests instead of a single interest in a very specific domains.

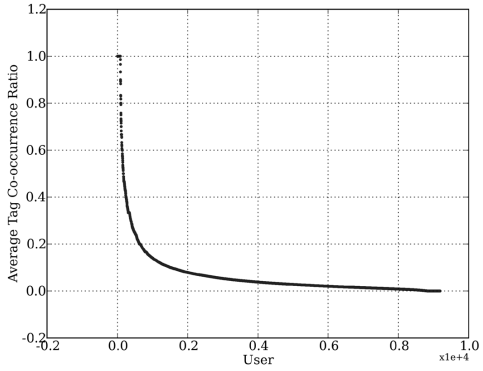
4. USER PROFILE CONSTRUCTION

As the majority of users in del.icio.us are observed to be interested in a wide range of topics from different domains, a user profile in the form of a single set of tags is definitely inadequate. Hence, user profiles which can accommodate the multiple interests of the users are very much desirable.

Identifying the different interests can be a challenging task



(a) Tag utilisation



(b) Average tag co-occurrence ratio

Figure 3: Distribution of tag utilisation and average tag co-occurrence. ratio

as tags are freely chosen by users and their actual meaning is usually not very clear. A solution to this problem is to exploit the associations between tags and documents in a folksonomy. As it is obvious that documents related to the same interest of a user would be tagged by similar tags, we can perform clustering algorithms on the documents tagged by a user to group documents of similar topics together, and extract the sets of tags assigned to these documents as indicators of the users' different interests.

Based on this idea, we propose a method for constructing user profiles which involves constructing a network of documents out of a personomy, applying community-discovery algorithms to divide the nodes into clusters, and extracting sets of tags which act as signatures of the clusters to reflect the interests of the users.

4.1 Community Discovery Algorithms

Clusters in a network are basically groups of nodes in which nodes have more connections among each other than with nodes in other clusters. The task of discovering clusters of nodes in a network is usually referred to as the problem of discovering community structures within networks [6]. Approaches to this problem generally fall into one of the two categories, namely agglomerative, which start from isolated

nodes and group nodes which are similar or close to each other, and divisive, which operate by continuously dividing the network into smaller clusters [20].

To quantitatively measure the 'goodness' of the clusters discovered, the measure of ***modularity*** [17] is usually used. The modularity of a particular division of a network is calculated based on the differences between the actual number of edges within a community in the division and the expected number of such edges if they were placed at random. Hence, discovering the underlying community structure in a network becomes a process to optimise the value of modularity over all possible divisions of the network.

Although modularity provides a quantitative method to determine how good a certain division of a network is, brute force search of the optimal value of modularity is not always possible due to the complexity of the networks and the large number of possible divisions. Several heuristics have been proposed for optimizing modularity, these include simulated annealing [10], and removing edges based on edge betweenness [17]. In addition, a faster agglomerative greedy algorithm for optimizing modularity, in which edges which contribute the most to the overall modularity are added one after another, has been proposed [16]. In this paper, we will employ this fast greedy algorithm to perform clustering, as it is efficient and performs well on large networks.

4.2 Construction of User Profiles

Given a network of documents (which are bookmarks in our case), we can apply the community-discovery algorithms to obtain clusters of documents. As the different clusters should contain documents which are related to similar topics, a cluster can be considered as corresponding to one of the many interests of the user. A common way to represent user interests is to construct a set of tags or a tag vector. Similarly, we can obtain a set of most frequently used tags from each of the document clusters to represent the corresponding interest. As a summary of our method, the following list describes the whole process of constructing a user profile for user u.

1. Extract the personomy $\mathbf{P}_u$ of user u from the folksonomy $\mathbf{F}$, and construct the bipartite graph TD_u.
2. Construct a one-mode network of documents out of TD_u, and perform modularity optimization over the network of documents using the fast greedy algorithm.
3. For each of the clusters (communities) c_i obtained in the final division of the network, obtained a set K_i of tags which appear on more than f% of the documents in the cluster. The set of tags of a cluster is treated as a signature of that cluster.
4. Finally, return a user profile P_u in the form of a set of $K_i's$: $P_u = \{K_i\}$.

For the signatures of the clusters, one can include all the tags which are used on the bookmarks in the cluster, or include only the tags which are common to the bookmarks in the cluster. However, the set of tags chosen for a cluster will affect how accurate the profile is in modelling the user's interest. In general, for a large value of f only the most common tags in the cluster will be included in the signature, while a small value of f will include more tags in the signature. We will investigate the problem of choosing a right value for f in the following section. As an illustrating example, Table 4 shows the results of applying the proposed method on two personomies, with $f = 20\%$.

User A	
K_1	webdesign, web2.0, tutorial, blog, css
K_2	linux, opensource, ubuntu, software
K_3	webhosting, filesharing
K_4	grammar, english
K_5	digg, sharing, music, mp3

User B	
K_1	webdesign, programming
K_2	interesting, art, video, funny
K_3	food, books, tobuy
K_4	lort, debate

Table 4: User profile constructed for two users.

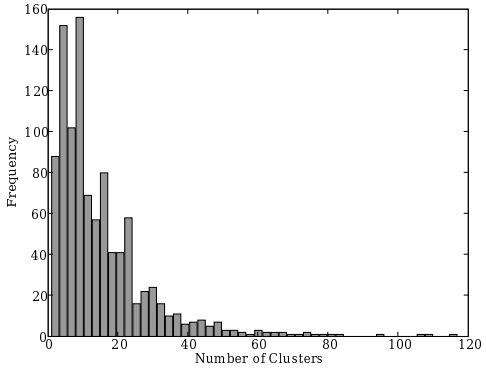


Figure 4: Number of clusters discovered for the 1000 personomies.

5. EVALUATION AND DISCUSSIONS

From our data set, we select at random 1,000 users who have over 100 bookmarks in their personomies. The requirement of having at least 100 bookmarks is to ensure that there are enough bookmarks for clustering so that clearer results can be obtained. We apply our proposed method of generating user profiles on these personomies, and obtain a set of clusters of bookmarks and their signatures. We discover that there are a substantial number of clusters with only one bookmark. The bookmarks in these clusters are mostly not assigned any tags. Hence, we exclude these single-bookmark clusters in the following analysis. Figure 4 graphs the number of clusters discovered for each of the personomies. On average 15 clusters are discovered in each personomy.

We believe that the use of multiple sets of tags in user profiles should give a more accurate representation of the interests of the users. Therefore we try to evaluate our proposed method by asking the following question: are the sets of tags accurate descriptions of the clusters of bookmarks from which they are extracted? If this is the case, then the user profiles should accurately represent the interests of the users. In the following we present the evaluations which

attempt to answer this question.

5.1 Precision and Recall Measures

Our question concerns with the issue of whether the sets of tags in the user profile are accurate descriptions of the bookmarks in the clusters. An appropriate method of evaluation is to approach this question from an information retrieval perspective. Given the signature of a cluster as a query, can we retrieve all the bookmarks within that cluster and avoid obtaining bookmarks in other clusters which are irrelevant? In addition, how many tags should be included in the signature in order to accurately described a cluster? To answer such questions, we will employ the measures of precision and recall [23] which are commonly used for evaluating information retrieval systems.

Precision and recall are two widely used measures for evaluating performance of information retrieval. Precision measures the fraction of documents in the retrieved set which are relevant to the query, while recall measures the fraction of relevant documents that the system is able to retrieve.

To employ the precision-recall measures, we treat the signatures of the clusters as queries, and use them to retrieve bookmarks by comparing the tags assigned to them to those in the queries. As for the representation of tags, we employ a vector space model of information retrieval. In other words, for each personomy, we construct a term vector $\vec{e} = (e_1, e_2, ..., e_n)$ for each bookmark, with $e_i = 1$ if the bookmark is assigned the ith tag, and $e_i = 0$ otherwise. Similarly, the signature of a cluster is converted into a query in the form of a term vector $\vec{q}$. The retrieval process is carried out by calculating the cosine similarity between the query vector and the bookmark vectors:

$$Sim(\vec{q}, \vec{e}) = \frac{\vec{q} \cdot \vec{e}}{|\vec{q}||\vec{e}|} \quad (4)$$

Those with similarity higher than a certain threshold t will be retrieved ($0 \leq t \leq 1$). For a cluster c, let the set of bookmarks in the cluster be D_c, and the set of bookmarks retrieved by the signature of the cluster be D_x. The precision and recall of the system on c are defined as follows. In addition, we also consider the F_1 measure [23] which is a combined measure of precision and recall.

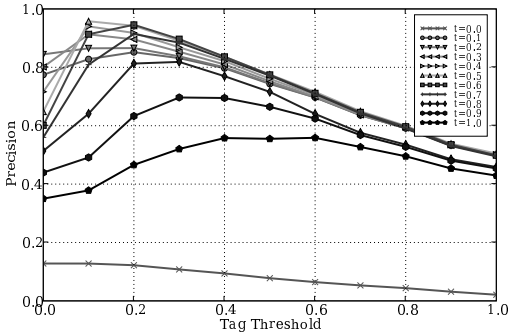
$$Precision(c) = \frac{|D_x \cap D_c|}{|D_x|} \quad (5)$$

$$Recall(c) = \frac{|D_x \cap D_c|}{|D_c|} \quad (6)$$

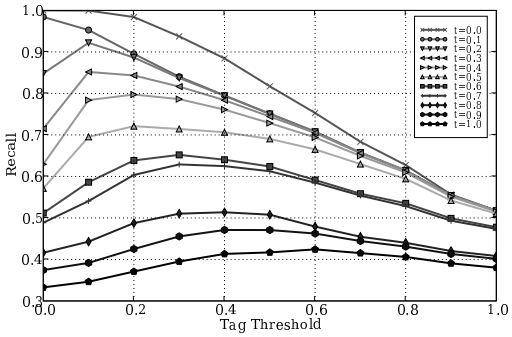
$$F_1(c) = \frac{2 \times Precision(c) \times Recall(c)}{Precision(c) + Recall(c)} \quad (7)$$

We calculated the three measures for the user profiles generated from the 1,000 selected personomies. We control two parameters in our evaluation, one is the value of f, the percentage of bookmarks above which a tag is assigned to in a cluster for it to be included in the signature, and the value of t, the threshold of cosine similarity. The results are presented in Figure 5.

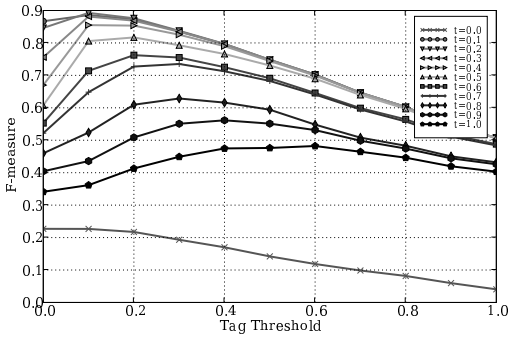
Figure 5(a) shows that for most values of similarity threshold precision attains maximum for f in the range from 0.1 to 0.4, and thereafter it continues to decrease as f increases. The result suggests that if only the most common tags are included in the signatures, they will become less representative as summaries of the clusters. This is probably due to the fact that the most common tags are usually too general



(a) Precision



(b) Recall



(c) F_1 measure

Figure 5: Precision, recall and F_1 measure. Different lines correspond to different values of similarity threshold.

and a query constructed from these tags will tend to retrieve bookmarks from other clusters as well which are related to a different sub-topic under the common tags. On the other hand, when one includes all the tags which appear in a cluster (with $f = 0\%$), the signature will include too many tags such that it will not be similar to any of the signatures of the bookmarks, leading again to a low precision.

As for recall, we observe some differences for different values of similarity threshold. For small values of t (from 0.0 to 0.3), recall continues to decrease as f increases. However, for larger values of t (from 0.4 to 1.0), recall first increases and

then decreases as t increases. This is probably due to the reason that when the similarity threshold is low, the number of tags in the cluster signature is less important as most of the bookmarks will be retrieved even if their similarity with the query is small. As f increases, fewer tags are included in the signature and therefore it becomes more difficult to retrieve relevant bookmarks. On the other hand, when t becomes higher, signatures which include all the tags in a cluster or include only the most common tags are very dissimilar to any of the bookmarks in the cluster, therefore recall attains maximum somewhere between the two extremes.

For common values of similarity threshold between $t = 0.3$ to $t = 0.5$, precision and recall attain maximum for values of f between 0.1 and 0.2, with precision over 0.8 and recalls over 0.7. F_1 measures also attain maximum around these values of t and f. This suggests that it is better to include more tags in a cluster signature so as to make it specific enough for representing the topic of the cluster (and thus the interest of the user represented by the cluster). Given these results, we conclude that by choosing a suitable value of f the tags extracted do constitute good descriptions of the bookmarks within the clusters.

5.2 Potential Applications

Our proposed algorithm provides a new way for constructing better user profiles based on the data available from collaborative tagging. There are a number of areas in which such algorithms can be applied to. We briefly discuss two of them in this section.

Firstly, as the user profiles provide a summary of the different interests of the users, it can be readily used to facilitate the management and organization of personal Web resources. For example, the sets of tags representing the clusters of bookmarks can be used to facilitate navigation and retrieval of a user's own bookmarks in del.icio.us. This would be much more efficient than navigating through the bookmarks by a single tag.

In addition, the user profiles can also be used to support Web page recommendation systems. Currently, del.icio.us provides various methods which allow users to keep track of new bookmarks which they may find interesting, such as subscribing to the RSS feed of a tag, or adding a user of similar interests to one's network. However, there have been no mechanisms which directly recommend interesting bookmarks to the users. With the user profiles constructed by our proposed method, recommendation systems will have a better understanding of the interests of the users, and be able to recommend more specific bookmarks to users by targeting a particular interest of the users.

6. RELATED WORK

User profile representation and construction has been a key research area in the context of personal information agents and recommendation systems. The representation of user profiles concerns with how user interests and preferences are modelled in a structured way. Probably the simplest form of user profile is a term vector indicating which terms are interested by the user. The weights in the vector is usually determined by the *tf-idf* weighting scheme as terms are extracted from documents interested by the user or obtained by observing user behaviour [2, 12]. More sophisticated representations such as the use of a weighted network of n-grams [21] have also been proposed. However, a single user profile vector may not be enough when users have multiple interests in diverse areas [7], and several projects have employed multiple vectors to represent a user profile. For example, Pon et al. [19] use multiple profile vectors to represent user interests to assist recommendation of news articles. In recent years, user-profiling approaches utilizing the knowledge contained in ontologies have been proposed. In these approaches, a user profile is represented in terms of the concepts that the user is interested in an ontology. For example, Middleton et al. [14] propose two experimental systems in which user profiles are represented in terms of a research paper topic ontology. Similar approaches have also been proposed to construct user profiles for assisting Web searching [26] or enhancing recommendations from collaborative filtering systems [1].

On the other hand, since the rise in popularity of collaborative tagging systems, some studies have also focused on generating user profiles from folksonomies. For example, in [5] a user profile generator based on the annotations assigned by the users to the documents is proposed. The user profile is represented in the form of a tag vector, which each element in the vector indicating the number of times a tag has been assigned to a document by the user. In [13], three different methods for constructing user profiles out of folksonomy data have been proposed. The first and simplest approach is to select the top k mostly used tags by a user as his profile. The second approach involves constructing a weighted network of co-occurrence of tags and selecting the top k pairs of tags which are connected by the edges with largest weights. The third method is an adaptive approach called the *Add-A-Tag* algorithm, which takes into account the time-based nature of tagging by reducing the weights on edges connecting two tags as time passes. In addition, [18] discusses the issue of constructing a user profile from a folksonomy in the context of personalised Web search. In their approach, a user profile p_u is represented in the form of a weighted vector with m components (corresponding to the m tags used by the user). The use of $\boldsymbol{w}_d$ is to assign a weight between 0 and 1 to each of the n documents. While these attempts provide some possible methods for constructing user profiles based on data in folksonomies, the possibility of a user having multiple interests is not addressed in these works.

7. CONCLUSIONS

The emergence of collaborative tagging systems provide valuable sources of information for understanding user interests and constructing better user profiles. In this paper, we investigated the characteristics of personomies extracted from folksonomies, and observed that the majority of users possess a wide range of interests, which cannot be modelled by simple methods such as a single set of tags. A novel method for constructing user profiles which take into account the diversity of interests of the users is proposed. We also evaluated the user profiles by looking at whether they provide a good summary of the bookmarks of the users.

This research work provides insight into how user profiles of multiple interests can be constructed based on the data collected from a folksonomy. From this point, we plan to carry out further research work in two main directions. Firstly, we will further investigate how the proposed method can be improved. In our study, a user profiles constructed treats every cluster of bookmarks and its signature as cor-

responding to a distinctive interest of the user. However, it may be true that two interests are related and are only subtopics of a more general area. We will investigate if the introduction of a hierarchical structure is desirable. Secondly, we will attempt to evaluate our proposed method by applying the user profiles on applications such as Web page recommendation or personal resource management. We hope this research will ultimately deliver useful algorithms and applications which utilise the power of user-contributed metadata in collaborative tagging systems.

8. REFERENCES

[1] Sarabjot Singh Anand, Patricia Kearney, and Mary Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Trans. Inter. Tech.*, 7(4):22, 2007.

[2] Marko Balabanovic and Yoav Shoham. Learning information retrieval agents: Experiments with automated web browsing. In *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogenous, Distributed Resources*, pages 13–18, 1995.

[3] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[4] Paul Alexandru Chirita, Andrei Damian, Wolfgang Nejdl, and Wolf Siberski. Search strategies for scientific collaboration networks. In *P2PIR '05: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*, pages 33–40, New York, NY, USA, 2005. ACM Press.

[5] Jörg Diederich and Tereza Iofciu. Finding communities of practice from user profiles based on folksonomies. In *Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice*, 2006.

[6] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PROC.NATL.ACAD.SCI.USA*, 99:7821, 2002.

[7] Daniela Godoy and Analia Amandi. User profiling in personal information agents: a survey. *Knowl. Eng. Rev.*, 20(4):329–361, 2005.

[8] Scott Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[9] Miha Grcar, Dunja Mladenić, and Marko Grobelnik. User profiling for interest-focused browsing history. In *SIKDD 2005 at Multiconference IS 2005*, Ljubljana, Slovenia, 2005.

[10] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895, 2005.

[11] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNCS*, pages 411–426. Springer, June 2006.

[12] Henry Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[13] Elke Michlmayr and Steve Cayzer. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, 16th International World Wide Web Conference (WWW2007)*, May 2007.

[14] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[15] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semant.*, 5(1):5–15, 2007.

[16] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[17] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[18] Michael Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, pages 365–378, November 2007.

[19] Raymond K. Pon, Alfonso F. Cardenas, David Buttler, and Terence Critchlow. Tracking multiple topics for finding interesting articles. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–569, New York, NY, USA, 2007. ACM.

[20] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *PROC.NATL.ACAD.SCI.USA*, 101:2658, 2004.

[21] H. Sorensen and M. Mcelligot. Psun: A profiling system for usenet news. In *CKIM'95 Workshop on Intelligent Information Agents*, 1995.

[22] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2004. ACM Press.

[23] C. J. van Rijsbergen. *Information Retrieval*. Dept. of Computer Science, University of Glasgow, 1979.

[24] Thomas Vander Wal. Folksonomy definition and wikipedia. *http://www.vanderwal.net/random/entrysel.php?blog=1750*, November 2, 2005. Accessed 13 Feb 2008.

[25] S. Wasserman and K. Faust. *Social network analysis*. Cambridge University Press, Cambridge, 1994.

[26] Xujuan Zhou, Sheng-Tang Wu, Yuefeng Li, Yue Xu, Raymond Y. K. Lau, and Peter D. Bruza. Utilizing search intent in topic ontology-based user profile for web mining. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 558–564, Washington, DC, USA, 2006. IEEE Computer Society.

Setting Access Permission through Transitive Relationship in Web-based Social Networks

Dan Hong Vincent Y. Shen
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong
{csdhong,shen}@cse.ust.hk

ABSTRACT

The rising popularity of Web 2.0, such as blogs, forums, online calendars/diaries, etc., makes users more interested in keeping their data on the Web. Sharing of such data could make life more enjoyable and convenient. For example, posting new photos about activities or sharing views about an event can let friends know what a user cares about. However, some of these data (such as a person's location during a particular time, opinion about a political event, etc.) are private and should not be accessed by unauthorized users. Although Web 2.0 facilitates sharing, the fear of forwarding sensitive data to a third party without knowledge of the data owners discourages people from using certain applications due to privacy concerns. We take advantage of the existing relationships on social networks and build a "trust network" with transitive relationship to allow data sharing while respecting the privacy of data owners. The trust network linking private data owners, private data requesters, and intermediary users is a directed weighted graph. The permission value for each private data requester is automatically assigned in this network based on the transitive relationship. Experiments were conducted to confirm the feasibility of constructing the trust network from existing social networks, and to assess the appropriateness of permission value assignments in the query process. This privacy scheme can make private data sharing manageable by data owners, who only need to define the access rights of their closest contacts once.

Categories and Subject Descriptors

K.4.1 [**Computers and society**]: Public Policy Issues—*Privacy*; H.3.5 [**Information storage and retrieval**]: Online Information Services—*Data sharing*; H.1.2 [**Models and principles**]: User/Machine Systems—*Human factors*

General Terms

Design, Human Factors

Keywords

Data Privacy, Trust Network, Transitive Relationship, P3P, Social Network

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.

1. INTRODUCTION

With the increasing popularity of Web 2.0 services, more and more Web users post their articles, pictures, comments on the Web through blogs, forums or other Web applications. Based on the "State of the Blogosphere" report [19], 120,000 new weblogs are being created worldwide each day. Many online communities have been established when users create accounts at those website hosts. In these communities users share their beliefs, opinions and interests. Communities on these websites are set up based on the "common interest" page their members marked. Each person can be members of several different communities and private data (e.g. identification, financial record, location, calendar, Web content) are commonly shared along community connections. However, these data sharing activities through Web-based social networking bring serious privacy concerns since users do not have control over who can access their personal data.

Nowadays, many users use a Web-based calendar, such as Google Calendar [9], to arrange their appointment schedules. It is possible to provide a feature to let users define activity categories, such as "family activity", "work activity", "church activity", etc. Figure 1(a) shows such a calendar which has several different categories. When a visitor of the website clicks on an item of the calendar, detailed information (such as location, contact person, etc.) about the event is displayed. It is also possible to provide a feature for the owner to define different groups (user context) who may access different categories of the calendar. For example, assuming Alice is the owner of such a calendar. As a family member, her sister Karen can see "family activity" in detail but not the detailed information of events in other categories. This strict definition of groups is useful, but it does not fully satisfy Alice's needs. To make the calendar more useful, some undefined visitors should also be allowed to see part of her calendar. Consider the following two scenarios:

- Bob, who is one of Alice's colleagues, can check her schedule and see the details of her "work activity". Carl, who is Bob's friend, hopes to make an appointment with Alice for some business discussion.
- Donald, who is Alice's travel agent, can check her schedule for the arrangement of a family vacation. Edward, who works for the car rental company which is a business partner of Donald's agency, needs the information regarding the family's arrival time.

The normal action for Carl is to ask his friend Bob to make the appointment for him. He may also write to Alice directly. This requires some amount of interactions between Carl and Bob, and may also involve Alice directly or indirectly. It will be more convenient if Carl can inherit some access right from Bob, who is Alice's

(a) Without Privacy Management (b) With Privacy Management

Figure 1: Web Calendar Example

colleague, and can check Alice's calendar directly for her "work activity" items when he visits her website. Donald (the travel agent) and Edward (the car rental agent) are in a similar situation; they should have the right to see the "family activity" category, but not the "work activity" category. Moreover, Alice's calendar can be checked by Donald and Edward based on additional context: Alice might only allow Donald to check her calendar after the final arrangement of her trip is settled and before the end of her trip (time context); and Edward is only allowed to check Alice's calendar information related to Edward's city (location context) and during the trip (time context, inherited from Donald).

It is hard for Alice to assign a special group and access right to every potential user for different calendar categories. It is not possible to assign an access right to someone whom Alice does not even know, such as Carl and Edward. But a "trust network" can be used to derive specific access rights when needed. The network is a directed graph which represents the trust relationship among users in it. During the query process, some *private data owners* (PDOs) might be willing to share their private data with *private data requesters* (PDRs) through the network. We note that the trust relationship is transitive; i.e., Alice trusts Bob and Bob trusts Carl implies Alice trusts Carl to a certain extent. It is also directional; i.e., although Alice trusts Carl by implication, Carl may not trust Alice regarding his private data. Since the trusted PDR through transitive trust relationship might have less access right (Edward does not have the same right as Donald has in the above example), the information released to indirectly-trusted PDRs may need to be obfuscated according to the level of trust. The trust network therefore requires:

1. Trust relationship defined by PDOs

2. Obfuscation (Web data annotation) rules defined according to the nature of private data

With the help of obfuscation rules, the access right is no longer binary ("yes" or "no"). The access right for a private data item is considered a PERMISSION VALUE, which represents how much detail the private data item can be given to the PDR based on the level of trust. Figure 1(b) shows the result when Carl looks at Alice's calendar when he visits the website. From the figure we can find out that Carl can only see the "work activity" and for the "family activity" Carl only knows that Alice is busy. The ability to control the sharing of private data makes life easier since Carl does not need to ask Bob, who is Alice's colleague, to help checking Alice's calendar.

In this paper, we are not focusing on how to define Web data in various levels of obfuscations. We solve the problem of assigning data access permission values when there is an existing social network. The contributions of this paper include the construction of a trust network from existing social networks. This network can be used to manage the sharing of private data in the Web environment. This trust network concept may be applied to data sharing in other ubiquitous computing environments.

The rest of the paper is organized as follows. Section 2 describes the related effort in improving privacy management. Section 3 describes how to bootstrap the trust network from an existing social network. Using the Web calendar as a case study, Section 4 demonstrates the process of trust network initialization and data sharing with obfuscation rules. A framework on how the components of the system to manage private data sharing can be implemented is given in Section 5. Section 6 summarizes the experiments we have done using an existing social network (MSN.com) to study the characteristics and significant issues of the trust network. Section 7 discusses possible refinements for the permission assignment techniques. Section 8 contains the conclusions and future work.

2. RELATED WORK

In order to identify Web users and their relationship with others, the Friend of a Friend (FOAF) [2] project creates a set of machine-readable pages describing people, the links between them, and the things they create and do. This could be the basis to construct trust networks by bootstrapping from existing social networks.

Much of the fundamental work in the analysis of social networks and the major advances in the past century have been carried out in the fields of sociology, psychology, and communications [8, 22]. The first step to facilitate social networking is to have a definition of trust that captures the social features for both local and global scopes [24]. Trust management is quite well studied in P2P systems and semantic Web [13, 14, 16, 23, 24]. In [14], a definition that captures the nature of social trust relationships and an algorithm are proposed for computing the trust value in social networks using default logic. Kamvar *et al.* proposed EigenTrust for reputation management for file sharing in P2P systems [13]. Richardon proposed a trust value computation method using probability theory in global belief combination which can provide each user a personalized set of trust values [16]. Trust propagation is another important research topic. Guha *et al.* proposed a method for predicting trust between users [10]. The trust acquisition and propagation model is discussed in [5, 6, 25]. However, the relationship between trust and online private data is not well addressed.

The online data privacy problem has been noticed for quite a long time. The Platform for Privacy Protection (P3P) Project [21] of the World Wide Web Consortium (W3C) is a method for websites to publish their privacy policies. The APPEL language [20]

works with P3P and enables users to exchange privacy preferences according to published privacy policies. P3P has not yet received much acceptance from Web users mainly due to its lack of enforcement, since current implementations do not include compliance of user preferences. Kolari *et al.* have pointed out that an enhanced P3P based on the Rei language can provide an improved trust model [15].

A lot of research has also been done on statistical databases to protect privacy through query restriction, data perturbation and output perturbation [1]. Such research focuses on hiding the relationship between the identity of the PDO and relevant private data [11, 18]. An example is that instead of giving the application an exact location, a regional context is used to satisfy the K-anonymity requirement. A list of candidates is returned to obfuscate private data [17]. There has not been much attempt to connect this approach with access control rules.

Since P3P does not provide any mechanism to ensure that these promises are consistent with internal data processing at the website, a purpose-based access control method can be used as an extension of P3P [4]. To address this issue we have proposed to extend the P3P protocol, which is a W3C recommendation for Web applications. We have successfully applied this extension to some context-aware applications [12]. But the extension did not consider transitive trust relationships. PDOs still need to specify every potential PDR's access right based on the categories defined by P3P, which makes management of private data cumbersome.

3. FROM SOCIAL NETWORK TO TRUST NETWORK

In the Web-based social network, the PDOs need to have some control on the management of their private data. However, it is not practical for a PDO to set a particular permission value for each private data category for every potential PDR. The role-based access control (RBAC) has partially solved the problem [7]. In this approach, it is required to define all the potential users' into some groups. For example, in the UNIX file system, the file owner (user) can give each role (user, group, and other users) some specific permissions (read, write, execute). With the role-based access control, a PDO needs to define the permissions based on the roles of PDRs. But it still may be difficult to define the role of every potential PDR. Therefore it will be very nice if a transitive trust relationship exists among the potential PDRs.

It turns out that the transitive relationship does exist in our daily life. For example, if Carl wants to know how much is the toll to travel through the Cross Harbor Tunnel in Hong Kong, he may ask his friend Bob about it. If Bob does not have the answer, he may continue asking his friends by phone calls or by emails. Later from Alice, Bob finds out the toll charge and passes the information to Carl. Formally speaking, this transitive query continues until a satisfactory answer is obtained and returned to the originator along the query path.

When each person who is willing to share data in the community is represented by a vertex, and when how much a PDO trusts a PDR is represented by an edge, the whole community becomes a trust network. When users share private data in a community, the access decision is based on the trust relationship between the PDO and the PDR in the trust network.

DEFINITION 1. *The* TRUST *relationship between a PDO and one of its contacts is a permission value assigned by PDO to a potential PDR:*

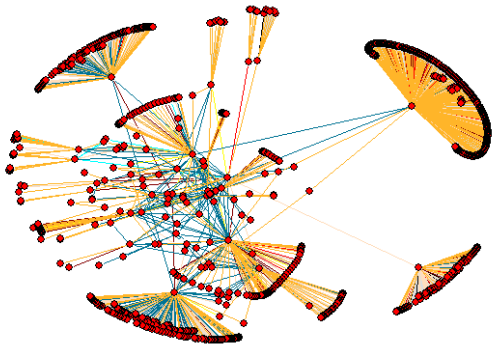


Figure 2: Facebook Network Example (1190 nodes). The data includes friends of ID 655183482 and friends of friends.

$$permission = trust(a, g, c)$$

Where a is the PDO involved, g is a member within a group of contacts that the PDO has defined, and c is the context where the permission value applies. The context in Definition 1 provides the application developer and the PDOs the ability to set the constrains in data sharing. Context may be related to the time, location, nature, etc. of an event. For example, Alice only allows Bob to view his calendar on her "working" activity. The event type "working" in the calendar can be considered as one context. It is extensible based on the needs of the application or the PDOs.

PDOs are requested to define data access permissions for all the direct users using their privacy preferences. The permission value can be a decimal number ranging from [0,1], where 1 represents total trust and 0 represents no trust at all. The 0 permission value is seldom used in online social networks because a PDO joins the network for the purpose of sharing data with friends there. The context in Definition 1 refers to the particular situation a permission value is assigned. The context includes time context, location context, and query context (such as purpose, retention, etc.) When Web data annotation is available in the social network, the annotation can also be part of the query context. For each kind of private data, the PDO can define several permission values to fit different contexts. A GROUP represents a group of PDOs who share the same permission value. A group can either be defined by a third party or by a PDO. One of the most popular Web-based social networks, Facebook, allows users to create private groups or to join the existing regional or alumni networks. Figure 2 shows "my friends" and "friends of my friends" relationship on Facebook for one of the authors. We can see that the relationship has been defined between Facebook users through the profile. When a PDO assigns his friends the permission which can be written in a preference file, the network becomes the trust network. The preference file can be stored as a single document or attached to the private FOAF document [2]. The trust relationship described above only supports the direct relationship. In the Web calendar application, the transitive trust relationship also needs to be considered. Carl, who is not directly connected with Alice, links to Alice through Alice's colleague Bob. In order to achieve this, we define a new operation JOIN.

DEFINITION 2. TRANSITIVITY *determines whether a trust relationship can be extended outside of the directly-connected PDRs. A propagated trust (Ptrust) relationship based on transitivity can be used to extend the relationship to other users. The* JOIN *opera-*

tion shows that the trust relationship is transitive; that is, if PDO A trusts PDR B, who in turn trusts PDR C, it implies that PDO A Ptrusts PDR C .

$$\forall a : \text{PDO}, i, j : \text{GROUP}, c : \text{CONTEXT}, \exists interim \in i$$
$$trust[a, i, c] = p_1, trust[interim, j, c] = p_2 \Rightarrow$$
$$Ptrust[a, j, c] = trust[a, i, c] \bowtie trust[interim, j, c] = min(p_1, p_2)$$

With the JOIN operation, the permission propagates along the trust network with the maximum possible value. Every potential PDR can be assigned a permission value automatically if he is within the community or from a related community. In a real application a PDO might set more restricted access. Additional operations will be proposed in the future.

4. TRUST NETWORK AND OBFUSCATION

Privacy management is separated into four steps: context pruning, transitive trust network initialization, permission value computation and data obfuscation. The four steps are applied when appropriate. In this section, we use the example when Edward sends a query on Alice's "family activities" in the calendar application to demonstrate these four steps.

4.1 Context Pruning

In a Web-based social network, users are allowed to define a lot of relationships with other users. For example, in Facebook users can define every relationship with all friends, such as "We went to school together" or "We took the course together". Moreover, a user can further specify which school and which course to establish the link between two users. As a result, group definition is quite complicated. Each PDO might need to define permission values for an individual person or a group based on different contexts.

The goal for context pruning is that trust relationship only propagates within the same group of people. For example, Alice would like to share her "work activity" with Bob. But she may not wish to share the information with Bob's family doctor, whom Bob trusts totally. Therefore the trust network should be restricted by context. We zoom in Figure 2 and extract part of the real Facebook network as shown in Figure 3. The church events in the calendar can be exchanged among all members of this network since all these five people are from the same church "CBIBC". But the work event is just shared between Michelle and Cammy since they "worked together" and no other user in the network has a similar context. Here "church" or "work", which might be an attribute of the event, can be considered a context.

Suppose there are two groups of users trusted by a PDO and a PDR is in both of the groups. If the PDR requests information from the PDO then it might be reasonable for the PDO to provide the larger permission value derived from each of the two groups. Another task for context pruning is to find out the maximum permission value for every direct trust relationship on the condition of satisfying the context requirement.

In the previous example, during the trip time the travel agent Donald is trusted by Alice based on RECIPIENT "ours"(see the definition in [21]). For other times, since the query context is not satisfied, Donald is not trusted.

4.2 Transitive Trust Network Initialization

Even if a PDO defines only a small portion of the whole community, data can still be shared based on the PDO's preferences. The users a PDO trusts may also have their own trust relationships (e.g., Donald trusts Edward due to partnership). We need to merge all the

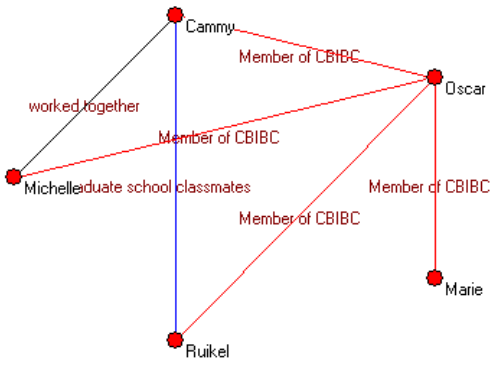


Figure 3: FaceBook Multiple Relationship Example.

relationships together to build the trust network. For the example discussed in 1, after context pruning we know the direct trust relationships form a tree. Figure 4 shows the result after merging all direct trust relationship trees of Alice, Bob, Edward and Donald.

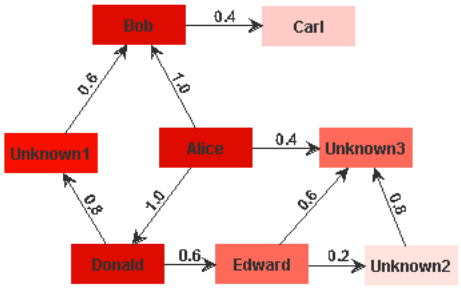


Figure 4: Trust Network- Transitive Relationship

DEFINITION 3. *In a trust network, the hops of a PDR is the number of vertices to traverse along the shortest path from the PDO to this PDR.*

Even if the complexity of privacy preference files has been decreased by using group-based permission assignments, to define the permission values of every potential PDR is still plenty of work. With the transitive relationship, a PDO only needs to define the permission values of those PDRs who have a "close" relationship, or are directly connected in the trust network. Based on the privacy preferences defined for each of these PDRs, the trust relationship can be computed and propagated to the rest of the trust network. Since there are various types of private data on the Web, we need to consider the data categories, sharing contexts during the trust network merging process.

4.3 Permission Value Computation

Note that to apply the transitive relationship, all the trust relationships during the propagation process need to have the same context. Before computation of the permission value for a PDR, context pruning will ensure the network initialized in 4.2 is extendable.

Algorithm 1 can be applied to implement the JOIN operation in order to compute the shortest path from the PDO (source) to a PDR

(destination). Given a social network graph $G(V, E)$, where V is the vertices set and E is the trust relationship set. $p(u, v)$ is user v's permission value given by user u. $Extract\_MAX(Q)$ is used to extract the vertex with the maximum permission value which is not in the finished set S. Through Algorithm 1, a user can get the most private data from a PDO based on the permission value assigned. Algorithm 1 is only one simple and possible solution to compute the permission value. The pageRank [3] or Max-Flow might be used to defined and compute the Ptrust.

Algorithm 1 Permission Value Computation

Input:A weighted directed graph G(V,E)
edge weight, p(u,v), is the permission from u to v
PDO, PDR
Output: Permission Value

```
 1: for all vertex v in V do
 2:    permission[v]=0
 3:    previous[v]=undefined
 4: end for
 5: permission[PDO]= 1
 6: S= empty set
 7: Q= V[G]
 8: while Q is not an empty set do
 9:    u= Extract_MAX(Q)
10:    if u equals PDR then
11:       return permission[u]
12:    end if
13:    S= S union u
14:    for all edge (u,v) outgoing from u do
15:       if min(permission[u], p(u,v)) > permission[v]
          then
16:          permission[v] = min(permission[u], p(u,v))
17:          previous[v] = u
18:       end if
19:    end for
20: end while
```

When Algorithm 1 is applied to Figure 4, it first puts Donald and Bob into the waiting queue Q. Then the $Extract\_MAX$ function extracts Donald from the queue and puts Edward and Unknown1 into Q. Then Bob is extracted and Carl is put into Q, too. The $Extract\_MAX$ function processes Unknown1 and Edward in order. When Edward is handled, the algorithm knows Edward's permission value. Therefore the trust is propagated from Alice to Donald and finally to Edward. We compute the permission value of every potential user (all users except Alice herself), and use a gradient color to represent the value as shown in Figure 4. The darker the vertex's color, the higher permission value it holds. We can see the effects of trust propagation by the changing color shades.

4.4 Data Obfuscation

There are lots of data items that can be represented in a hierarchical way. For example, the "current location" is a frequently-used private data in different applications. Room 4208, Floor 4, HKUST, Hong Kong, China is a common address to define a location precisely. To protect privacy, for some PDRs in some applications, a PDO may want different information shown on the PDR's screen. Detailed information (room number, etc.) is given to close friends and general information (Hong Kong) is given to unknown PDRs. Based on the transitive relationship, the permission value can be used to control the degree of obfuscation for a certain private data item based on either the default value or the user's preference.

From Figure 4, we see that when the trust network becomes complex it is quite possible for an unknown PDR to obtain private data after several data passing actions. In order to make sure the private data passing scale is controllable, a PDO needs to set up some control factors:

1. Maximum propagation hops, hop_{max}: how many hops private data can be passed along the network. This is helpful to stop data propagation to PDRs who are too far away.

2. Damping factor, ϖ: How much data is obfuscated through every hop. This method gradually reduces the information available and makes sure that an unknown PDR cannot get too much detailed information through several trustable intermediary users.

Therefore we can replace line 15-18 of Algorithm 1 by:

```
if min(permission[u], p(u,v)) × ϖ > permission[v]
hop[v] ≤ hop_max ∧ u is not PDO then
   permission[v] = min(permission[u], p(u,v)) × ϖ
   previous[v] = u
end if
```

With the help of hop_{max} and the damping factor ϖ, the private data is controlled to spread only within a certain number of hops. Moreover, the farther a PDR is away from a PDO, the less private data he receives. For the previous example, Edward can know Alice is in HKUST without the damping factor. And if the $\varpi = 0.7$, then permission for Edward is 0.42. Edward can only know that Alice is in Hong Kong. The permission values might be hard for PDO to understand. It is helpful to visualize the social network by painting users in the network with colors of different shades based on the permission values assigned as shown in Figure 8. And it is also very helpful to assign the critical person, who has lots of connection the PDO are not familiar with, a sharp ϖ in order to keep the data private.

5. FRAMEWORK OVERVIEW

In the Web calendar example, we use the Privacy Server framework as shown in Figure 5. The PDOs define their private data through the PDO Preference Manager and store their preferences in the PDO Preference Database. When there is a data query initiated by a PDR, the Private Data Query Adapter acts as an interpreter for the query and sets up the trust network based on PDO's preference definition. The data query should include all the context information (e.g., the reason to access the data, how to forward data to third parties and application user name). With the PDO information from Context Database, the Adapter computes the permission value based on PDO's preference and passes the value to the Obfuscation Manager. A fuzzy result is returned based on applicable obfuscation rules and the permission value.

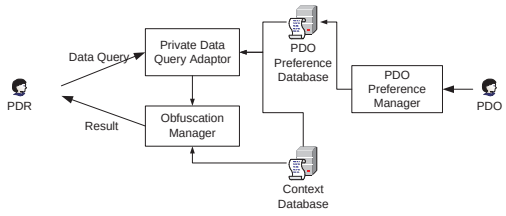


Figure 5: Privacy Server Implementation Framework

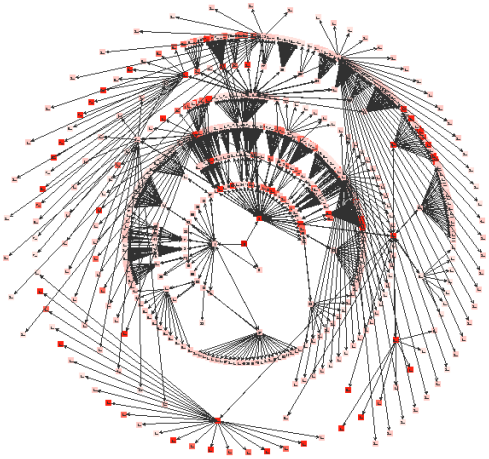
A trust network is set up based on the transitive relationship defined by each PDO, which is derived from the online community

information. Users in a whole community who are willing to share private data become vertices in the network while the trust relationship between each other becomes edges. The strength of the trust relationship becomes private date permission value which denotes the edge weight in the trust network. Since the trust relationship is asymmetric, the whole trust network is a directed graph. When there is a private data query, the problem becomes the checking of whether there is a path from a vertex (PDO) to another vertex (PDR) in a directed graph.

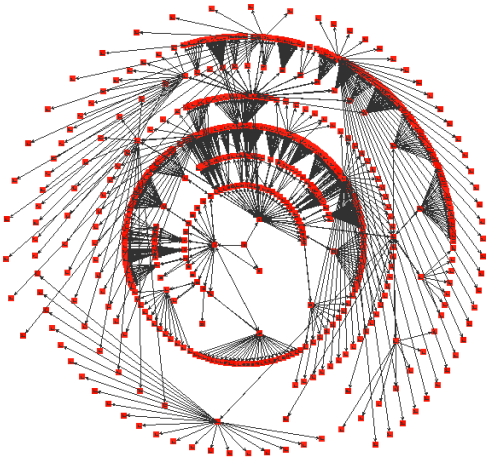
After the permission value is obtained, the Obfuscation Manager blurs the private data according to the value and still returns some information to the PDR (unless the permission value is zero, indicating that the PDR is forbidden from accessing the data). Context data can often be represented in many ways and forms. For example, the location context can be represented at a particular point geographically, or in regions of various sizes which contain that point. Alice's location, in the previous example, could be represented as <Alice, at, Cross Harbor Tunnel, Hong Kong, China>, showing that Alice's location information at a certain time is one of Cross Harbor Tunnel, Hong Kong and China depending on the permission value. The Obfuscation Manager returns different results for different queries based on the relationship between the PDO and the PDR.

The transitive relationship and obfuscation rules break the current binary private data access characteristic and make context sharing easier. We modify the Web calendar component, JEvent, and build the Privacy Management Framework for it as shown in Figure 1. Figure 1(a) is the original JEvent service. Users are allowed to check all the detailed calendar information by clicking on the event. With the privacy management as shown in Figure 1(b) only the registered users can check the calendar and the "Family activities" is not available based on the data category the calendar owner (PDO) has defined. The successful hacking of the code for JEvent shows that the transitive trust relationship does work in a real application.

This framework is not specially defined for the Web calendar; other applications can also connect to the Privacy Server through an HTTP connection for the current CGI version.



(a) Random Permission Value Assignment



(b) Assign High Permission value for Relationship

Figure 6: Transitive Network Efficiency

6. EXPERIMENT

6.1 General Characteristics of the Trust Network

Our study is focused on the "trust network" where edge (u, v) means u trusts v with a labeled permission value. There are lots of online communities available currently, such as MSN, Facebook, Blogger, etc. We picked MSN due to its popularity to test the implementation of permission value assignment scheme. Starting from one of the authors' friends who posts her friends list on the Web[1], we used a crawler to trace the friends lists. We visited 187 users who are connected with the friend within four hops and obtained other 1,181 related users. None is more than four hops away from the friend. Since there was no permission value currently supported by MSN, we randomly assigned different permission values for every relationship.

Figure 6 contains the trust propagation results after we randomly assigned permission values using Math.random (range [0,1)). The permission values became very small after four hops as shown in Figure 6(a), since most peripheral nodes are in light color. If these peripheral nodes wish to see the central node's information, their requests will not be successful. Since friend lists on MSN are defined by the users manually, the trust relationships should be higher than random assignments in the range of [0,1). By changing the range to [0.6,1), the results are shown in Figure 6(b). Compare Figure 6(a) and Figure 6(b), we see that the colors in Figure 6(b) are darker, which means that the permission values are higher after trust propagation when higher permission values are assigned initially. Therefore the permission values defined by PDOs are indeed affecting the private data propagation process.

[1]MSN URL:http://rp20040619.spaces.live/friends

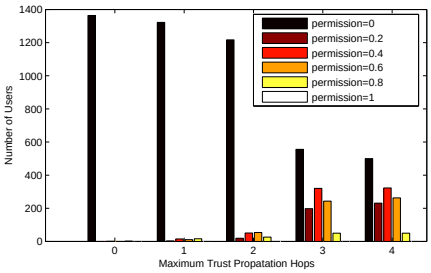
6.2 Control Factors

Figure 6 demonstrates that it is possible to construct a trust network from an existing social network for managed data sharing, if the social network supports the setting of permission levels. We then explore how a PDO can control the transitive relationship with partial trust.

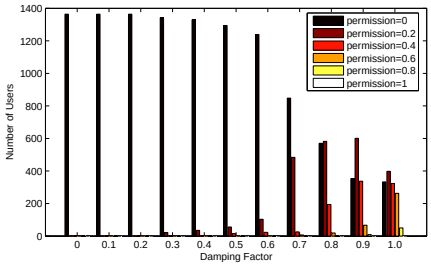
The maximum number of hops hop_{max} can be set by a PDO in order to control how far the private data can be forwarded. We again use the MSN social network as a test base. We randomly assigned permission values to every trust relationship and then kept this directed graph unchanged in the following experiment.

When no transitive relationship was allowed ($hop_{max} = 0$), 1,350 queries got no permission during data sharing in Figure 7(a). When the transitive relationship was allowed, the non-empty query number was dramatically increased when $hop_{max} = 3$. This is

because there are few users on the first one or two hops of the trust network. The bigger hop_{max} was, the more detailed result could be obtained. Moreover, we noted that blanket permission was not granted since only a small number of queries could get access to the private data. We can also see that even if the friend has only defined three close friends, if she allows three hops of data sharing, then around 700 users can see her obfuscated data.



(a) Max Hop Number Affects Permission



(b) Damping Factor Affects Permission

Figure 7: Control Factors

Figure 7(b) demonstrates how the damping factor discussed in section 4.4 affected the permission value. If the damping factor ϖ is zero, it meant that there was no transitive relationship. If ϖ is very small (e.g., 0.1 or 0.2), it strongly restricted the access permission of private data. Even when ϖ became 0.6, most users got permission value less than 0.2. When ϖ became bigger, the influence of ϖ significantly affected the permission value to access private data.

We understand that the number of users who get permission might be different due to different social network topologies. For example, if the PDO defines a lot of close friends, there will be a number of users who get permission to access private data even when $hop_{max} = 0$,. The selection of ϖ and hop_{max} will indeed affect the topology of the trust network. But the trend of trust propagation will not change too much. In practice the damping factor should be used with maximum hop number together in order to achieve the desired access control. Moreover, the PDO can set up different damping factors to different groups or specific users if he wishes.

7. DISCUSSIONS

7.1 Trust Priority

In a trust network, it is possible that a PDR may obtain more private data through transitive relationships. For example Figure 8 is the result after running Algorithm 1. The gray line represents the trust propagation path when Unknown3 queries Alice's information. Through a full transitive relationship, Unknown3 can get 0.6 permission value through the path: Alice $\rightharpoonup$ Donald $\rightharpoonup$ Edward $\rightharpoonup$ Unknown3. However, Unknown3 is directly defined in Alice's trust tree with permission value 0.4 (green line). There is now a conflict between the direct trust and trust derived from transitive relationships. Since trust based on multiple recommendations from a single source should not be higher than that from independent sources, if the PDR is one of the directly-connected vertices with the PDO then the permission for this PDR cannot be higher than the permission value originally assigned by the PDO.

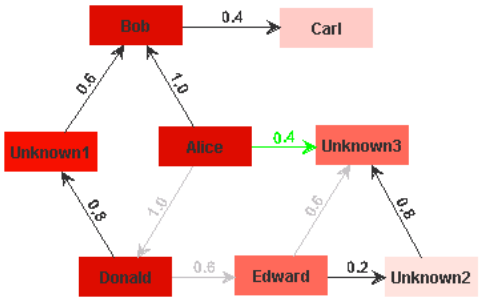
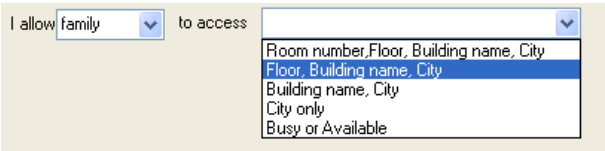


Figure 8: Trust Priority: Directed vs. Transitive

7.2 Standardized Private Data Levels

It is often hard for users to assign accurate decimal permission values to others. Therefore, we should provide a visualization of the data. The user can directly select the data level they would like to share and the program can easily convert the level into a decimal number.



The levels of a private data item are either defined by a PDO or by a public ontology. Then different PDOs might have different data levels in real applications. For example, Alice defines her location in 5 levels, such as "Room 4208, Floor 4, HKUST, Hong Kong, China". Her secretary only uses "HKUST, Hong Kong, China". When the secretary grants Bob with permission value 0.67, Bob can only know "Hong Kong". If Alice gives a 0.8 permission value to her secretary, then with the transitive relationship Bob gets 0.67 (the maximum of 0.8 and 0.67) permission value and consequently a more specific area name (HKUST) of Alice's location. This could be a big privacy hole.

A possible solution is that for each category a standardized private context data level is set up and shared by all PDOs through a separate central information directory which provides all kinds of information level descriptions. The PDO Preference Manager connects to that directory and automatically helps users to search what other preferences the PDO has defined. Initially, there are only a few default levels for the data. When a PDO wants to have more specific context levels, he can insert a level himself and record the new level in the central information directory. For example, if Alice wants to identify the current building as a new context level, she finds that this information is between the floor information and the area name. Alice can then insert the building name between them and set the permission value for this new context level to be $(\frac{0.6+0.8}{2} = 0.7)$. When other PDOs define their location information, this new level can also be used by them. Since the permission

value is a decimal number between 0 and 1, an infinite number of context levels can be supported. Another advantage of using standard levels is that a PDO can see and directly choose the information level he wishes to share with other users instead of assigning a permission value which may not be meaningful to the PDO.

7.3 Other Applications

With the development of ubiquitous computing, more and more private data is available to the public either on the Web or through other applications. For example, a mobile service provider has already started friend location service. Users can dial a special number to trace friends' location. Users might lose privacy control in that situation because he may not know what information about him is shared, compared with the social network situation that the user is the publisher of his own data on the Web. It is possible that through such a service, a thief can find out a user's regular schedule, such as the time to go home, by tracking the user's location for a period of time before breaking into his home when he is not there. The convenience of ubiquitous computing applications will not be enjoyed unless users can control what private data to share with whom at what time. Our privacy server framework can be helpful in these applications.

8. CONCLUSIONS AND FUTURE WORK

In this paper we propose a transitive trust network for private data sharing in social networks. Private information can be shared through the trust network. We use a Web calendar application to show the process of using trust network algorithms to share data. We finally demonstrate the feasibility of constructing the trust network from an existing social network. The characteristics of such a trust network are analyzed which may be applied to data sharing in ubiquitous computing environments. We plan to launch the Web calendar service with trust network and collect data for further development. We shall also develop plug-ins and propose to owners of social networks that users be allowed to use them to assign permission values to their contacts.

Acknowledgement

This research is supported in part by Hong Kong Research Grant Council Project 619507.

9. REFERENCES

[1] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Survey*, 21(4):515–556, 1989.

[2] D. Brickley and L. Miller. Foaf project. http://www.foaf-project.org/, 2007.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[4] J.-W. Byun, E. Bertino, and N. Li. Purpose based access control of complex data for privacy protection. In *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110, New York, NY, USA, 2005. ACM Press.

[5] M. Conrad, T. French, W. Huang, and C. Maple. A lightweight model of trust propagation in a multi-client network environment: To what extent does experience matter? In *ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*, pages 482–487. IEEE Computer Society, 2006.

[6] B. Esfandiari and S. Chandrasekharan. On how agents make friends: mechanisms for trust acquisition. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies 2001*, pages 27–34, 2001.

[7] D. F. Ferraiolo, J. F. Barkley, and D. R. Kuhn. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security*, 2(1):34–64, 1999.

[8] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3(1), June 1997.

[9] Google. Google calendar. www.google.com/calendar.

[10] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM Press.

[11] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *VLDB*, pages 9–16, 2006.

[12] D. Hong, M. Yuan, and V. Y. Shen. Dynamic privacy management: a plug-in service for the middleware in pervasive computing. In *MobileHCI 2005*, pages 1–8, Salzburg, Austria, September 2005. ACM.

[13] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[14] Y. Katz and J. Golbeck. Using social network-based trust for default reasoning on the web. Submitted to Journal of Web Semantics, 2007.

[15] P. Kolari, L. Ding, S. G. A. Joshi, T. Finin, and L. Kagal. Enhancing web privacy protection through declarative policies. In *POLICY '05: Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, pages 57–66, Washington, DC, USA, 2005. IEEE Computer Society.

[16] R. Matthew, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, 2003.

[17] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, pages 763–774, 2006.

[18] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[19] Technorati. State of the blogosphere / state of the live web. http://www.sifry.com/stateoftheliveweb/, 2007.

[20] W3C. A p3p preference exchange language 1.0 (appel1.0). http://www.w3.org/TR/P3P-preferences/.

[21] W3C. Platform for privacy preferences (p3p) project. http://www.w3.org/TR/P3P/, April 2002.

[22] S. Wasserman and K. Faust. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1994.

[23] C.-N. Ziegler and G. Lausen. Analyzing correlation between trust and user similarity in online communities. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proceedings of the 2nd International Conference on Trust Management*, volume 2995 of *LNCS*, pages 251–265, Oxford, UK, March 2004. Springer-Verlag.

[24] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, pages 83–97, Washington, DC, USA, 2004. IEEE Computer Society.

[25] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, pages 83–97, Washington, DC, USA, 2004. IEEE Computer Society.

Tagpedia: a semantic reference to describe and search for Web resources

Francesco Ronzano
Institute for Informatics and Telematics (IIT) - CNR
Via Moruzzi, 1
Pisa, Italy
francesco.ronzano@iit.cnr.it

Andrea Marchetti
Institute for Informatics and Telematics (IIT) - CNR
Via Moruzzi, 1
Pisa, Italy
andrea.marchetti@cnr.it

Maurizio Tesconi
Institute for Informatics and Telematics (IIT) - CNR
Via Moruzzi, 1
Pisa, Italy
maurizio.tesconi@iit.cnr.it

ABSTRACT

Nowadays the Web represents a growing collection of an enormous amount of contents where the need for better ways to find and organize the available data is becoming a fundamental issue, in order to deal with information overload. Keyword based Web searches are actually the preferred mean to seek for contents related to a specific topic. Search engines and collaborative tagging systems make possible the search for information thanks to the association of descriptive keywords to Web resources. All of them show problems of inconsistency and consequent reduction of recall and precision of searches, due to polysemy, synonymy and in general all the different lexical forms that can be used to refer to a particular meaning. A possible way to face or at least reduce these problems is represented by the introduction of semantics to characterize the contents of Web resources: each resource is described by one or more concepts instead of simple and often ambiguous keywords. To support these task the availability of a global semantic resource of reference is fundamental. On the basis of our past experience with the semantic tagging of Web resources and the SemKey Project, we are developing Tagpedia, a general-domain "encyclopedia" of tags, semantically structured for generating semantic descriptions of contents over the Web, created by mining Wikipedia. In this paper, starting from an analysis of the weak points of non-semantic keyword based Web searches, we introduce our idea of semantic characterization of Web resources describing the structure and organization of Tagpedia. We introduce our first realization of Tagpedia, suggesting all the possible improvements that can be carried out in order to exploit its full potential.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous; D.2 [**Software**]: Software Engineering

General Terms

semantic resource, knowledge organization, semantic web

Keywords

semantics, web, social, wikipedia, data mining

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.

1. INTRODUCTION: KEYWORD BASED SEARCHES

Currently, keyword based Web searches are the preferred way to seek for resources of interest over the Web. Each resource, usually identified by its URL, can be accessed by one or more keywords describing its content. The most widespread methods to explore links between Web resources and keywords are **the exploitation of a search engine** or **the access to a collaborative tagging service** (see Figure 1).

Search engines like *Google*, *Yahoo*, *Ask* and so on are examples of automated information extraction systems: they analyze the data and the structure of Web contents as well as the search behaviour of users and the frequency of usage of different search strings to collect the most appropriate keywords that can be used to access a Web resource (see the lower portion of Figure 1).

On the other side, collaborative tagging systems like *delicious*, *Flickr*, *YouTube* and *Technocrati* rely upon user contribution. They are examples of social classification systems: each person who belongs to the community of users of a collaborative tagging system describes Web resources of interest by means of one or more freely chosen keywords, called tags. All the tags associated to Web resources are collected and exploitable by every user in order to find many resources of interest. A popularity value is usually associated to each tag describing a Web resource to point out the number of times it has been chosen to characterize that resource and consequently the importance of the tag itself among those related to the specific resource (see the upper portion of Figure 1).

Even if they are very popular, **keyword based Web search approaches show many weak points in managing language expressivity**. Many keywords can identify distinct concepts (*polysemy*): as a consequence the precision of search results decreases. Moreover if we don't search for a common sense of that keyword, it is often very difficult to explore the search results space so as to find Web resources of interest among those retrieved. For example, let us suppose that we want to find all the resources dealing with 'ajax' intended as the Greek hero: choosing 'ajax' as search text string, there are no links related to mythology among the first 30 search results of Google. If we better specify the search string in order to solve the problem, we partition the space of relevant search results depending on the particular word added to 'ajax' to disambiguate its meaning. For instance, depending on the addition of the word 'hero' or the word 'mythology' to 'ajax' in the search string, considering

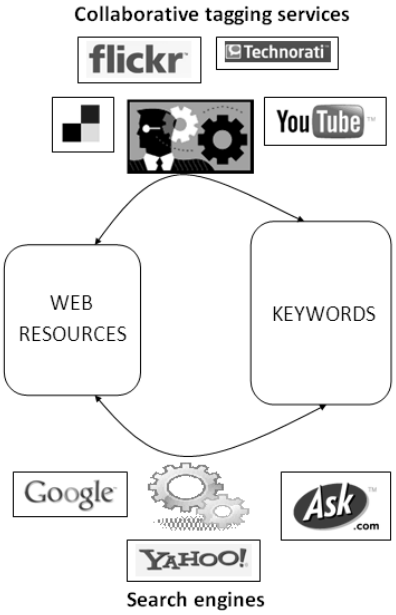


Figure 1: Two ways to associate keywords to resources

the first 10 search results shown by Google, only two of them are present in both cases. Besides polysemy, also *synonymy* affects precision and recall of keyword based Web searches. In fact, when a specific meaning can be accessed through two or more keywords, the set of search results is different depending on the particular keyword chosen. Moreover, the different *level of precision* and the *many possible users points of view* that can be considered describing a particular resource, often cause a considerable loss of quality of Web searches. For a deeper analysis of all the factors that affect efficiency and effectiveness of keyword based Web search systems see [4] [10] [12] [25].

In order to face the different drawbacks of the systems just analyzed, many distinct methods have been applied. The **aggregation of search results from different search engines and their post elaboration** is experimenting a growing diffusion. Systems like **Vivisimo** [15], **Grokker** [18] and **Kartoo** [19] are meta search engines. They collect search results from other search engines and group them exploiting, for example, the category hierarchy of Yahoo and Wikipedia (Grokker) or creating clusters of similar search results and characterizing each of them by one or more additional keywords (Vivisimo). They also display search results cartographically through very expressive maps that connect the most relevant resources to the most used keywords (Kartoo).

Also considering tagging systems, we can find many proposals to better organize search results to improve their quality and the effectiveness of the search. **FolkRank** [5] is an algorithm created to rank search results in a tagging system, calculating a ranking value for each of them and thus evaluating their relevance. Also user profile is exploited in order to adapt ranking calculation to the information needs of every single user.

The rest of this paper is organized as follows. In Section 2 we describe our idea of semantic characterization of Web resources, underlining the need for a general-domain semantic resource of reference in order to support this task, taking into account also our past experience with the semantic tagging and SemKey. In Section 3 we introduce Tagpedia, the semantic resource of reference we have created by mining Wikipedia, explaining its organization and structure (Subsection 3.1). In Section 4 we describe how Tagpedia can be utilized, describing the Tagpedia Web API and showing all the possible improvements to Tagpedia to exploit its full potential. Conclusions are described in Section 5.

2. FROM KEYWORDS TO CONCEPTS: SEMANTIC CHARACTERIZATION

We can solve, or at least substantially reduce, Web resources organization and classification problems by adding a further level of completeness in their characterization: **the semantics**. Instead of relying on post processing of search results, we can directly semantically describe resources thanks to their association with one or more properly chosen concepts. In this way we extend the characterization of resources introducing the semantic level: each resource (R) is described by one or more concepts (C) and in turn each concept can be accessed through one or more keywords (K) (see Figure 2). When we search for some informaton of interest, we can better specify our informative needs and we can easily and effectively access relevant results thanks to the support and the exploitation of the collection of concepts used to describe Web resources, referred to as semantic resource in what follows.

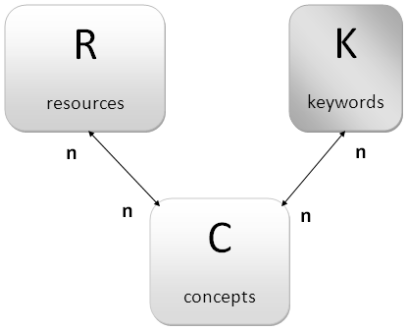


Figure 2: Relations between resources, keywords and concepts

This way of improving Web contents organization represents an attempt to *realize the semantic description of information that stands at the basis of the Semantic Web vision.*

At present there are many proposal of semantic classification methods for Web contents. **FolksAnnotation** [13], for instance, tries to extract the tags that describe a Web resource from a collaborative tagging system, automatically mapping them to the corresponding concepts of a predefined domain ontology. Such kind of systems usually require a strongly and well organized ontological frame of reference that is difficult to realize; they have not provided significant improvements in comparison with the classical keyword based methodologies. A different approach is those exploited by systems like **Semantic Halo** [3]: it improves tag based search systems adding semantic information without relying on ontologies. Analyzing co-occurrences and frequencies of

tags, Semantic Halo algorithm extracts groups of tags useful to better specify and drive user search, like more general or more specific ones or group of keywords defining a particular naming of the selected tag. Not enough experimental data on the effectiveness and usefulness of this method to improve tag based searches is currently available. Summarizing, *a strong and widespread infrastructure that organizes and provides access to Web resources on the basis of semantic classificatory information is still absent.*

During the first half of 2007, we have tried to realize the possibility to semantically describe Web resources developing SemKey [4], a semantic collaborative tagging system. It extends current tagging systems allowing to characterize resources by referring to concepts. Each user can point out and describe Web resources of interest: starting from a freely chosen tag, he can disambiguate it thanks to the support of *Wikipedia* [14] and *WordNet* [17] in order to identify one or more defined concepts. In this way he produces a semantic assertion that is the description of a specific feature of Web resources through one or more chosen concepts. Thus we can potentially overcome the limits in the description of Web resources related to the complexity of language, exploiting their semantic characterization as well as the semantic relations between concepts present in WordNet and Wikipedia.

We have implemented a working prototype of SemKey; by analyzing the usage patterns and the semantic classification support provided by our system, we have identified **two key factors that need to be improved in order to really make possible semantic characterization of Web resources**, as described in the previous part of this Section.

Both Wikipedia and WordNet, even if they show important features to support the semantic description of Web resources, are weakened by relevant lacks. WordNet presents a rich set of parts of speech and a strongly structured set of relations between them, but it lacks many data useful to support proper names disambiguation and it is not collaboratively edited. Wikipedia is an encyclopedia so its content is composed mainly by a very rich set of names along with their extended descriptions. Thus Wikipedia has strong proper names coverage and it has been proposed as a named entity disambiguation resource in [7] and [8]; it is also continuously updated, but lacks a structured set of relations between the concepts described, even if its documents are interconnected by a huge number of links and loosely classified through categories. As a consequence the semantic resources considered are in some way complementary, but they have been built and structured for purposes different from the semantic characterization over the Web. In order to better support this task **we need a semantic resource built and structured ad hoc**, which is still absent: it must feature all the advantages of those just analyzed, removing pointless informative contents.

Moreover, a great limit to the usability of SemKey and to an easy definition of new semantic metadata is represented by the different steps users must carry out to compose a semantic assertion. This often discourages them from creating semantic metadata. **Some sort of automation is necessary in order to speed up the tag disambiguation process or to execute it through automated procedures**.

3. TAGPEDIA: A GENERAL DOMAIN SEMANTIC RESOURCE OF REFERENCE

Starting from the need for a global semantic resource exploitable as a reference to describe Web contents and therefore comprehensive and updated, we have proposed a possible solution to this demand, designing and building Tagpedia. It is **a semantic organization and classification of tags, intended as words or in general brief textual expressions, that people may use to describe Web Resources**. Tagpedia is **based on the model of term-concept networks** [11], **structured ad hoc to support the semantic characterization of Web contents and initially populated exploiting Wikipedia data**. In particular we have tuned a new way of mining Wikipedia to extract the information needed to build Tagpedia so as to support concept based descriptions of Web resources also through tag disambiguation.

We have chosen **Wikipedia** as the starting point because **it represents the most rich and constantly updated encyclopedic reference over the Web with a huge set of semantic contents included, even if not explicitly exposed and easily accessible**. During the last few years many studies have been carried out finding new ways to extract useful semantic data exploiting the great amount of information contained in Wikipedia. Information organizational patterns like infoboxes, internal and external links, redirect and disambiguation pages have been analyzed in order to extract valuable data. The DBPedia Project [16], for instance, is a relevant attempt to extract semantic data from Wikipedia, making them available over the Web complying with Semantic Web standards [6]. DBPedia is a global knowledge base derived from Wikipedia, not specifically intended for Web resources description as Tagpedia is. In [24] there is a description of KLYN, a system that autonomously semantifies Wikipedia, automatically suggesting data inconsistencies, lacks or incompletenesses. Wikipedia has been also successfully exploited to compute semantic relatedness between words [21] and natural language texts [9], but also to tune new named entities disambiguation methodologies [7] [8]. Semantic relationships between Wikipedia categories have been studied in order to make the search of information easier and to give articles editors relevant suggestions [20]. Moreover some research has been done to understand and measure the way Wikipedia articles are created and their contents become mature [22] or to analyze statistical information about the growth of the data that constitute Wikipedia, the types of articles, the editors, the link and category structure and so on [23].

3.1 The structure of Tagpedia

The main aim of Tagpedia is the semantic characterization of data over the Web. In particular it must allow to describe a Web resource through the association with one or more univocally referenced concepts. Thus, **the main constitutional unit of TagPedia is the concept**. Each concept must be unequivocally identified but also easily accessed. The main way to point out a concept is through the words that refer to it. Such words will also be called tags in the following. As a consequence, each concept is identified by *the set of all the words or, more generally, all the alfanumeric expressions of any kind that can be adopted by a community of users to refer to it*, thus constituting a

set of synonymous tags or **syntag set**. Syntag sets are the molecules which form Tagpedia.

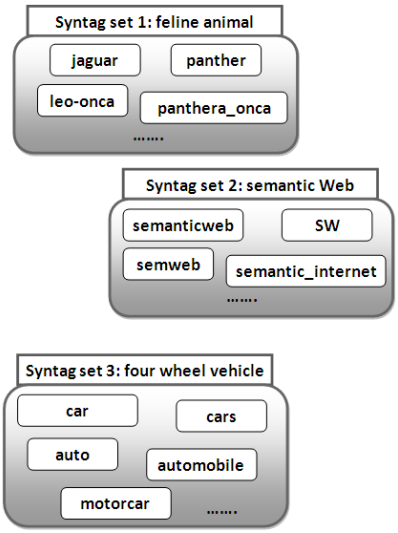


Figure 3: Three syntag sets

The creation of an initial rich collection of syntag sets is the first necessary step that must be carried out to build our semantic resource. Wikipedia shows many features exploitable to create such a collection of syntag sets. In particular, in Wikipedia an article usually defines a specific concept. As a consequence in order to bootstrap Tagpedia, we create syntag sets from the articles of Wikipedia. In Figure 3 we show three examples of syntag sets made up by tags collected mining Wikipedia.

To be more precise, Wikipedia pages can be substantially divided into three groups:

- **article pages**: each describes a particular concept, identified by the title of the same page;
- **redirect pages**: each links an alternate literal expression, that constitutes the title of the redirect page, to the corresponding concept, usually identified by the title of an article page;
- **disambiguation pages**: each lists all the possible concepts, usually identified through the titles of article or redirect pages, that can be referred by the literal expression constituting the title of the disambiguation page.

The redirect and the disambiguation page mechanisms are two important Wikipedia organizational solutions that can be exploited to build and enrich syntag sets.

Once identified a concept referring to a particular article page, we create an initial version of a syntag set, pointed out by a unique identifier, including only the tag corresponding to the title of the page (in Figure 3, considering the syntag set 1, the tag 'jaguar' is the title of an article page). Then we collect all the words and expressions that may be used to refer to that concept.

As previuosly mentioned, in Wikipedia the **redirect** mechanism is used to link alternate literal expressions to the original encyclopedic article that describes a specific entity. It is usually used to manage synonyms, abbreviations, acronyms, misspellings, other spellings, different punctuations, particular capitalization rules and so on. In TagPedia we mine Wikipedia content and extract all the redirect information analyzing redirect pages; for each of them we enrich the syntag set related to the referred concept by adding the title of the page as a new tag (in Figure 3, considering the syntag set 1, the tag 'leo onca' is extracted from a redirect page).

Moreover, Wikipedia usually manages polysemy through the **disambiguation pages**. As said, each disambiguation page represents a collection of links to all the different article pages that identify the distinct meanings pointed out by the page title (textual string). For example, the word '*ajax*' is highly polysemous and has 49 different meanings in Wikipedia: its disambiguation page contains links to 49 distinct article pages; each one identifies a particular concept. We analyze Wikipedia disambiguation pages as a futher source of information to enrich the syntag sets of Tagpedia through the addition of new words that refer to a defined meaning. In particular, for every disambiguation page, we point out each syntag set related to the concepts referenced inside its Wikipedia text and we add the title of the same disambiguation page as a new tag exploitable to access to the selected syntag sets (in Figure 3, considering the syntag set 1, the tag 'panther' is extracted from a disambiguation page).

Summarizing, let us define C_i a concept derived from a specific Wikipedia article page P_i. To populate with tags the syntag set for C_i we extract:

- the title of P_i;
- the title of every redirect page to P_i;
- the title of every disambiguation page containing a link to P_i.

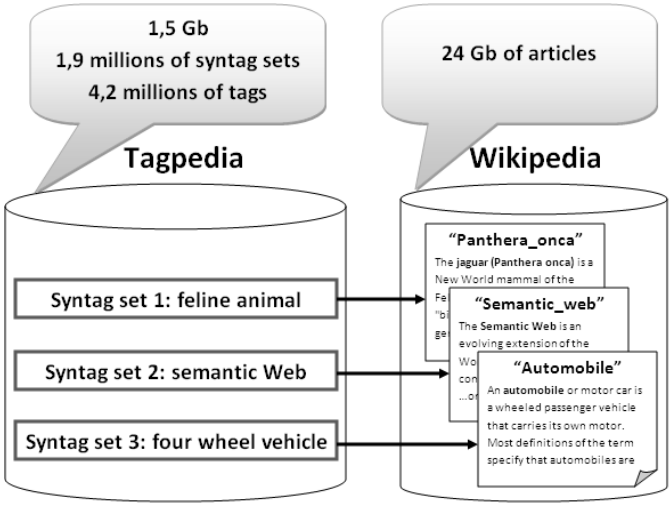


Figure 4: The structure of Tagpedia

Starting from a dump of the English version of Wikipedia, we have developed a set of C++ routines, that automatically analyze the text of Wikipedia articles. By mining structural

elements of Wikipedia syntax as well as by considering texts punctuation and by exploiting pattern matching techniques mainly based on regular expressions and string analysis, our routines gather all the concepts as well as all the possible tags used to refer to each single meaning, thus defining a huge collection of syntag sets. The meaning of each concept, identified by a syntag set, is also better specified by pointing to the corresponding article in Wikipedia.

All these data are collected in a relational database properly designed and optimized for a fast access. It is constituted by two basic collections: *the concept table* and *the tag table*. The first one gathers all the concepts of Tagpedia assigning to each of them a unique identifier, the Concept ID and a brief definition, extracted from the English version of Wikipedia. For every concept we also collect the URL of the corresponding Wikipedia article. On the other side, the tag table contains links between each concept, referenced through its identifier, and all the tags used to access to it.

By mining September 2007 dump of the English version of Wikipedia, we have obtained more than **1,9 millions of syntag sets** and more than **4 millions of tags** used to point out the intended concepts, each one referencing a specific Wikipedia article (see Figure 4).

Considering Figure 5, we can visualize the weight of the different sources of the 4.230.740 tags of Tagpedia. The number of tags extracted from article pages (P) is equal to the number of syntag sets, that is 1.927.378. Among the 2.303.362 remaining tags, 481.250 have been generated by mining disambiguation pages and 1.822.112 by analyzing redirect ones.

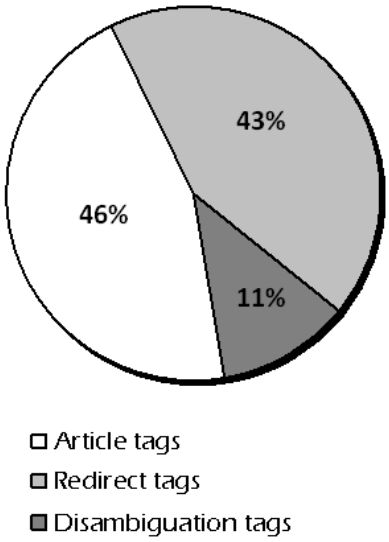


Figure 5: Sources of the tags in Tapgedia

This group of syntag sets constitutes the basis of Tagpedia providing a way to unequivocally access and refer to concepts when users must semantically describe or search for Web resources.

Number of del.icio.us URLs:	100	
Number of distinct tags:	1087	
Percentage of successfull disambiguations:	84	%

Table 1: Tagpedia tag disambiguation support: preliminary evaluation results

4. EXPLOITING AND IMPROVING TAGPEDIA

In order to support the generation of semantic descriptions of Web resources or to semantically search for Web contents, the information contained in Tagpedia should be easily accessed, querying the whole collection of syntag sets. For this purpose we have developed the **Tagpedia Web API** that is a simple set of procedures that may be invoked via Web to exploit the semantic support offered by Tagpedia. These procedures carry out few fundamental tasks and may be composed to realize more complex functions; their execution can be easily requested by other external Web applications so as to integrate semantic features.

The main tasks that Tagpedia Web API supports are:

- the definition of *all the possible meanings for a given tag*, i.e. all the syntag sets that contain the tag;
- the collection of *all the tags belonging to a specific syntag set*, i.e. all the words or expressions exploitable to access that particular meaning;
- the retrieval of *the short textual description of a specific syntag set*.

Exploiting Tagpedia Web API, we have integrated this semantic resource into SemKey, our semantic collaborative tagging system, substituting WordNet and Wikipedia so as to support the disambiguation of the meaning of tags. Once chosen one or more tags, the user specifies the right meaning for each of them, choosing a particular syntag set among those including the intended tag. An early prototypal Web-based interface useful to explore and interact with Tapgedia is accessible at the URL *www.tagpedia.org*.

In order to evaluate the coverage of Tagpedia and also to obtain suggestions to improve this semantic resource, we have tried to manually **point out the right meaning of the tags associated to the 100 most popular Web resources over del.icio.us**, tagged by more than 25000 users. Relying upon Tagpedia Web API, we have developed a Web based procedure that, starting from the URL of a Web resource retrieves all the related tags in del.icio.us. All the possible meanings of each tag are retrieved from Tagpedia along with their short descriptions and the user manually verify if the right concept is present. In this way, collecting all the results of our user based tests, we have obtained a first evaluation of the disambiguation effectiveness of our semantic resource. The results are shown in Table 1.

Tagpedia provides a valid support to the process of disambiguation for 84% of the total number of tags considered.

Anyway we have identified several different ways to improve its contents and, as a consequence, its semantic coverage and its usefulness. In the following part of this section we will describe these proposals for future works.

Despite its good disambiguation coverage, there are different particular tags like 'sem_web', 'inplaceedit', 'web_dev'

and similar ones that are not managed by Tagpedia, because they are non conventional words, often created by a user to describe a particular concept and then accepted and exploited by many others. One possible solution to this problem is **the introduction of collaborative Web editing techniques for Tagpedia contents**. Giving users the possibility to create new syntag sets or to merge or extend existing ones through new tags is fundamental for such a kind of resource. Indeed the effectiveness of Tagpedia in the description of Web resources is proportional to the possibility to adapt and enrich this semantic resource in respect to the variability of user descriptive needs. In this context, the introduction of the possibility to collaboratively collect and manage data, following a Wiki-like paradigm, represents a key factor of current Web and is a crucial issue considering Tagpedia.

Another aspect of Tagpedia that can be substantially improved is **the enrichment of its semantic contents with the addition of semantic relations between syntag sets**; they are useful to better identify concepts or to easily search for them. Each syntag set, representing a meaning, may be connected to other ones through relationships like specialization, generalization, relatedTo and similar ones. Possible ways to mine relevan relations between syntag sets are the analysis of the internal links between Wikipedia article pages as well as the exploitation of the hierarchy of Wikipedia categories. For instance, relying upon relations, when we specify the concept to search for or when we must choose a specific concept to semantically characterize a resource, the system can show the most general or the most specific ones to simplify this task. Similarly, during a semantic search, starting from a specific syntag set, if we can browse all the related ones, we can better specify our search needs and thus easily retrieve the desired information.

A third way to improve and enrich Tagpedia is **the definition of semi-automated procedures to extend its data, exploiting other resources and importing their contents into Tagpedia**. Other relevant free Web thesauri or dictionaries or other language tools can be valid sources of information. For instance the *Dictionary of Automotive Terms* [1] or the *Free Online Medical Dictionary* [2] are two domain specific resources that can be integrated in Tagpedia. Moreover, mapping rules between Tagpedia syntag sets and other Web semantic resources can be defined to integrate different sources of information thanks to the common gronud represented by Tagpedia itself.

Another aspect that must be further addressed in Tagpedia, is **the support for multilinguism**. In Tapgedia, each syntag set is language independent. The tags constituting that particular syntag set are specific to the particular language. Managing the possibility to collect different tags belonging to different languages into a syntag set, we can deal with different languages and once identified one or more particular concepts we can make language indipendent semantic searches. We think that this possibility should be better explored and defined, trying to determine specific semantic search patterns.

As already mentioned in the concluding part of Section 2, **the definition and tuning of automated or semi-automated procedures to create semantic descriptions** is a further important issue to be faced. Users should be allowed to semantically describe Web resources in an easy way; they must be supported in the task of turning simple keywords into concepts or browsing the collection of syntag sets constituting Tagpedia without complicating their usual interaction patterns or compromising the usability of the systems they interact with. Moreover, automated methodologies to derive semantic descriptions of Web resources from simple keyword based ones can also be tuned, so as to create an initial solid collection of semantic metadata and bootstrap this new way to characterize resources over the Web.

5. CONCLUSIONS

In this paper we have presented Tagpedia, a collection of tags semantically structured, built ad hoc to describe Web contents.

Starting from a brief analysis of the weak points of keyword based methodologies for information organization and searching and considering also the current approaches to face these issues, we have introduced the possibility to semantically describe Web resources through concepts. To make it possible, we have developed an initial version of Tagpedia a general domain semantic resource of reference, created by mining Wikipedia. After a description of its structure and organization and an overview of the Tagpedia Web API, useful to easily access and exploit the information collected in Tagpedia, we have focused our attention on the possible improvements to this semantic resource. Collaborative wiki authoring, syntag set relations enrichment, automated procedures for content extraction from external sources, support for multilinguism and automated generation of semantic descriptions of Web resources are some of the many improvements considered that can be carried out, underlining its broad enhancement possibilities.

On the base of all these considerations, we believe that Tagpedia, despite its initial stage of development, represents an important attempt to support the introduction of semantics over the Web, trying to put in practice the principles of the Semantic Web on a global scale and to better structure and manage the huge amount of data constituting the actual Web.

6. REFERENCES

[1] Dictionary of automotive terms. http://www.motorera.com/dictionary/.

[2] Free online medical dictionary. http://cancerweb.ncl.ac.uk/omd/.

[3] Alessio Malizia Alan Dix, Stefano Levialdi. Semantic halo for collaboration tagging systems. *In the Social Navigation and Community-Based Adaptation Technologies Workshop - June 20th, 2006, Dublin, Ireland.*

[4] Francesco Ronzano Marco Rosella Salvatore Minutoli Andrea Marchetti, Maurizio Tesconi. Semkey: A semantic collaborative tagging system. *In the Tagging and Metadata for Social Information Organization Workshop at the World Wide Web Conference 2007 - May 8, 2007, Banff, Alberta, Canada.*

[5] Christoph Schmitz Gerd Stumme Andreas Hotho, Robert Jĺaschke. Folkrank: A ranking algorithm for folksonomies http://www.kde.cs.uni-kassel.de. *In the Lernen - Wissensentdeckung - Adaptivität Workshop - October 9-11, 2006, Hildesheim, Germany.*

[6] Soren Auer and Jens Lehmann. What have innsbruck and leipzig in common? extracting semantics from

wiki content. *In the 4th European Semantic Web Conference - June 5th, 2007, Innsbruck, Austria.*

[7] Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. *In the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics - April 9-16, 2006, Trento, Italy.*

[8] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. *In the Empirical Methods in Natural Language Processing Conference - June 28-30, 2007, Prague, Czech Republic.*

[9] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. *In the Proceedings of the 20th International Joint Conference on Artificial Intelligence - January 6-12, 2007, Hyderabad, India.*

[10] Scott A. Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *In the Journal of Information Sciences, vol. 32, April, pag. 198-208, 2006.*

[11] Andrew Gregorowicz and Mark A. Kramer. Mining a large-scale term-concept network from wikipedia. *Mitre Technical Report, October 2006.*

[12] Marieke Guy and Emma Tonkin. Tidying up tags? *D-Lib Magazine*, 12, January 2006.

[13] Hugh C. Davis Hend S. Al-Khalifa and Lester Gilbert. Creating structure from disorder: using folksonomies to create semantic metadata. *In 3rd International Conference on Web Information Systems and Technologies - 3-6 March, 2007, Barcelona, Spain.*

[14] http://en.wikipedia.org/wiki/. The english version of wikipedia.

[15] http://vivisimo.com/. Vivisimo, search done right!

[16] http://wiki.dbpedia.org. Dbpedia.

[17] http://wordnet.princeton.edu/. Princeton wordnet.

[18] http://www.grokker.com/. Grokker enterprise search management.

[19] http://www.kartoo.com/. Kartoo meta-search engine.

[20] Wolfgang Nejdl Sergey Chernov, Tereza Iofciu and Xuan Zhou. Extracting semantic relationships between wikipedia categories. *In the 1st Workshop on Semantic Wikis at the 3rd European Semantic Web Conference - June 11-14, 2006, Budva, Montenegro.*

[21] Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. *In the Proceedings of the 45th Annual Southeast Regional Conference, pag. 106 - 110 - March 23-24, 2007, Winston-Salem, North Carolina, USA.*

[22] Cristopher Thomas and Amit P.Sheth. Semantic convergence of wikipedia articles. *In the Proceedings of Web Intelligence Conference, pag. 600-606 - Silicon Valley, November 2-5, 2007.*

[23] Jakob VoSS. Measuring wikipedia. *In the Proceedings of the 10 th International Conference of the International Society for Scientometrics and Informetrics - July 24-28, Stockholm, Sweden.*

[24] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. *In the Proceedings of the 16th ACM conference on Conference on information and knowledge management, pag. 41-50 - November 6-9, 2007, Lisboa, Portugal.*

[25] Jianchang Mao Zhichen Xu, Yun Fu and Difu Su. Towards the semantic web: Collaborative tag suggestions. *In the Proceedings of the Collaborative Web Tagging Workshop at the World Wide Web Conference 2006 - May 23-26, 2006, Edinburgh, Scotland.*

Hyperstructure Maintenance Costs in Large-scale Wikis

Philip Boulain
prb@ecs.soton.ac.uk

Nigel Shadbolt
nrs@ecs.soton.ac.uk

Nicholas Gibbins
nmg@ecs.soton.ac.uk

Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science, University of Southampton
Southampton SO17 1BJ, United Kingdom

ABSTRACT

Wiki systems have developed over the past years as lightweight, community-editable, web-based hypertext systems. With the emergence of Semantic Wikis, these collections of interlinked documents have also gained a dual role as ad-hoc RDF [8] graphs. However, their roots lie at the limited hypertext capabilities of the World Wide Web [1]: embedded links, without support for composite objects or transclusion.

In this paper, we present experimental evidence that hyperstructure changes, as opposed to content changes, form a substantial proportion of editing effort on a large-scale wiki. The experiment is set in the wider context of a study of how the technologies developed during decades of hypertext research may be applied to improve management of wiki document structure and, with semantic wikis, knowledge structure.

Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Information Storage and Retrieval—*Online Information Services*; H.5.3 [**Information Systems**]: Information Interfaces and Presentation—*Group and Organization Interfaces* ; H.5.4 [**Information Systems**]: Information Interfaces and Presentation—*Hypertext/Hypermedia*

General Terms

Experimentation

Keywords

Hypertext, Semantic Web, Wiki, Web Science

1. INTRODUCTION

This experiment forms part of a broader project looking into the potentially beneficial relationships between open hypermedia, the study of interconnected documents; Semantic Web, the study of interconnectable data; and 'wikis', web-based communal editing systems.

Hypermedia is a long-standing field of research into the ways in which documents can expand beyond the limitations of paper, generally in terms of greater cross-referencing and composition (reuse) capability. Bush's *As We May Think* [2] introduces the hypothetical early hypertext machine, the 'memex', and defines the "essential feature" of it as "the process of tying two items together". This *linking* between documents is the common feature of hypertext systems, upon which other improvements are built.

As well as simple binary (two endpoint) links, hypertext systems have been developed with features including n-ary links (multiple documents linked to multiple other documents), typed links (links which indicate something about *why* or *how* documents are related), generic links (links whose endpoints are determined by matching criteria of the document content, such as particular words), and composite documents, which are formed by combining a set of other, linked, documents. Open Hypermedia extends this with interoperation, both with other hypermedia systems and users, and with non-hypermedia resources. A key concept in open hypermedia is that of the *non-embedded* link—links (and anchors) which are held external to the documents they connect. These allow links to be made to immutable documents, and to be added and removed in sets, often termed 'linkbases'. One of the earliest projects attempting to implement globally-distributed hypertext was Xanadu [9], a distinctive feature of the design of which was *transclusion*: including (sections of) a document into another by reference.

In related work, we are currently investigating the relationship between an exemplar semantic wiki, Semantic MediaWiki [6], and open hypermedia systems, as defined by the Dexter Hypertext Reference Model [5]. Our preliminary results based on a formal description of Semantic MediaWiki in terms of the Dexter model suggest that such semantic wikis can be treated as simple open hypermedia systems. While details are beyond the scope of this paper, some basic parallels are evident: a wiki node is akin to a hypermedia document, and a semantic web resource. Semantic wikis generally treat typed inter-node links as RDF statements relating the nodes, and these links are embedded and binary in hypermedia terms. From this we can see a meaningful similarity between a graph of documents connected by typed links, and a graph of resources connected by RDF statements. We can also see that wikis do not have features covering more advanced hypermedia links: such as those which are not embedded, or have more than two endpoints.

This then suggests that semantic wikis stand to gain from techniques developed within hypermedia, but we must first judge if there is any substantial cost to be reduced. Hence we have performed an quantitative experiment on a large-scale public wiki system to measure the proportion of effort expended on hyperstructure-related activities, as opposed to editing the document content.

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.

2. HYPOTHESIS

We carried out an experiment to estimate the proportion of effort expended maintaining the infrastructure around data, rather than the data itself, on a weak hypertext wiki system. We define a 'weak' hypertext system here as one whose feature set is limited to embedded, unidirectional, binary links, as with the World Wide Web. Our hypothesis is that the manual editing of link structure, of a type which richer hypertext features could automate, will show to be a significant overhead versus changes to the text content.

This experiment also seeks to partially recreate a related, informal experiment, discussed in an essay by Swartz [10].

3. DATASET

We chose English Wikipedia[1] as the experimental dataset, because it has both a considerably large and varied set of documents, and a complete history of the editing processes—performed by a wide range of Web users—between their first and current versions[2]. The wiki community keep the dataset fairly well inter-linked and categorised for cross-reference, but they do this via the cumulative efforts of a large body of part-time editors. As well as being statistically significant, demonstrating possible improvement of English Wikipedia is socially significant, as it is a widely-used and active resource.

It is important to stress the size of the English Wikipedia dataset. Wikipedia make available 'dumps' of their database in an ad-hoc XML format; because this study is interested in the progression of page contents across revisions, it was necessary to use the largest of these dumps, containing both page full-text and history (unfortunately, also non-encyclopædic pages, such as discussions and user pages). This dump is provided compressed using the highly space-efficient (although time-complex) bzip2 algorithm; even then, it is 84.6GB. The total size of the XML file is estimated to be in the region of two terabytes.

4. PROCEDURE

Figure 1 shows the simplified data flow of the processing of the dump performed for the experiment.

First, we trimmed down the dataset to just those pages which are encyclopædic articles, as these are the pages of greatest significance to the Wikipedia project's goals, and thus the most important to study. Otherwise, the dataset would include a lot of 'noise' in the form of discussion and user pages, which are likely to have different editing patterns, and be less connected to the hyperstructure. The most practical way to do this was to remove any page placed in a namespace. On English Wikipedia, this also has the effect of removing other page types, such as media and image descriptions, help pages copied from MetaWiki, front-page portal components, and templates. As this stage also required decompressing the data, it ran over the course of several days on a multi-processor server.

We took a random subset of the data for processing. Samples of 0.04% and 0.01% of pages (approximately: see the description of the subset tool below; actual page counts 14,215 and 3,589 respectively) were selected, yielding a compressed dataset which would fit on a CD-ROM, and could be processed in a reasonable timeframe. Further iterations of the experiment may study larger subsets of the data.

We performed categorisation on the revisions, into several edit types which would be automatically distinguished. In particular, a simple equality comparison between a revision, and the revision two edits previous, can detect the most common (anti-)abuse modification: the rollback, or revert (unfortunately, MediaWiki does not record such operations semantically). A sequence of reverts[3] is usually indicative of an 'edit war', where two users continually undo each-others changes in favour of their own. Page blanking was also easy to detect, but identifying more complicated forms of vandalism (e.g. misinformation, spam) was not feasible—if reliable, automatic detection were possible, they would not be present in the data, as Wikipedia could prevent such changes from being applied. Identifying abuse (and abuse management) of the simpler types is important, as otherwise they would appear as very large changes.

In order to detect changes in the text content, templates used, MediaWiki categories, and links from a page, it was necessary to attempt to parse the MediaWiki markup format. Such 'wikitext', as it is known, is not a formally defined language: there is no grammar for it, and it does not appear likely that an unambiguous grammar actually exists. MediaWiki does not have a parser in the same way as processing tools such as compilers and XML libraries; instead it just has a long and complicated set of text substitution procedures which convert parts of 'wikitext' into display-oriented HTML. These substitutions often interact in a ill-defined manner, generally resulting in either more special-case substitutions, or as being defined as a new, hybrid, feature, which editors then use. Because of these problems, and the lack of abstraction in MediaWiki's 'parser', as much as the programming language boundary, a 'scraping' parser was created which attempted to approximate partial processing of the wikitext format and return *mostly* correct results. This parser is a single-pass state machine (42 states) with a few additional side-effects. This yields excellent performance: testing showed that the time spent parsing is dominated by the time performing decompression.

To determine if an edit included a significant ('major') change to the text content, we required a difference metric between the plaintext of the revisions. This metric was then compared to a threshold to classify edits as being content changes or not (in particular, the imperfect parser generates 'noise' from some non-content changes, as it cannot correctly remove all the markup). The default threshold was chosen as 5%: sentences in the English language are generally around twenty words in length, so this considers anything up to changing one word in each sentence as non-major (minor). MediaWiki also allows registered users to explicitly state than an edit is minor; this flag was respected where present.

We chose an approximation of Levenshtein distance[7], as it is a simple measure of insertions, deletions, and substitutions, fitting the kind of edit operations performed on the wiki. However, the algorithm for computing Levenshtein itself was far too time-complex, even with aggressive optimisation, taking two minutes on a tiny test set of just a few thousand revisions of a single page (before trimming away the identical parts at either end of both strings to take advantage of edit locality, this took 45 minutes). The problem

[1] `http://en.wikipedia.org/`

[2] MediaWiki, unlike many wikis, never deletes old revisions of a page.

[3] e.g. `http://en.wikipedia.org/w/index.php?title=Anarchism&diff=next&oldid=320139`

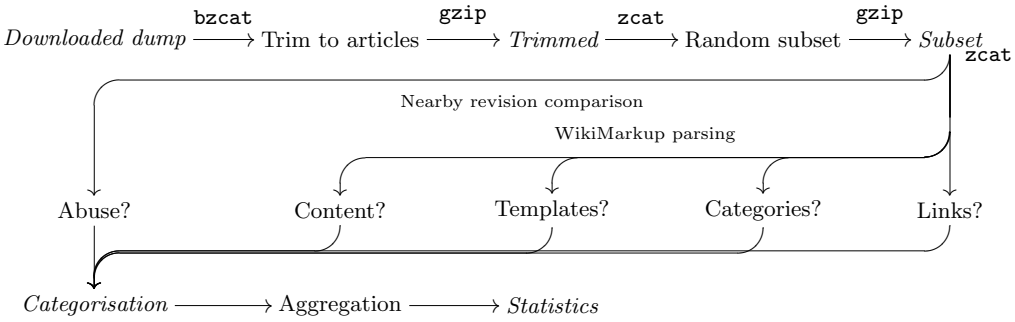


Figure 1: Data flow of Wikipedia experiment

was that the matrix-based approach is $O(n \times m)$, where n and m are the string lengths, in all cases: for n and m in the region of 16,000 characters, as found on many revisions, merely iterating through all 256 million matrix cells was prohibitively expensive.

Instead, we developed a new approach to computing such a distance, taking advantage of the domain-specific knowledge that the two strings being compared are likely very similar save for 'local' edits: the difference is likely to be a new paragraph, or a removed sentence, or some changed punctuation. Instead of efficient search within the space of editing operations, as Levenshtein, it is based on the idea of "sliding windows": a pass is made over both strings in parallel; when characters begin to differ, a look-back 'window' is opened between the point at which differences began, and continues until similarity is again found between these windows. At this point, the position through the strings resynchronises, the distance is increased by the offset required, and the windows are again 'closed'. When the end of either string is reached by the far edge of the window, the algorithm can terminate, as any remaining characters in the other string must be unmatched and thus add to the distance. As a result, the algorithm scales with regard to the shorter of the two strings, which is helpful when revisions may add whole paragraphs of new text to the end. To reduce inaccuracy in certain cases, the algorithm maintains a 'processed point' cursor, to avoid double-counting of overlapping insertions and deletions. Pseudocode is presented as algorithm 1, which works on a pair of string buffers, and `upstr.c` in the tool source contains a C implementation. This approach is still $O(n \times m)$ worst-case, but is $O(n)$ (where n is the shorter string) for identical strings, and degrades smoothly as *contiguous* differences increase in size: instead of two minutes, the tiny test set was compared in a little over ten seconds.

Unfortunately, changes such as 'ABCF' to 'ADCDBCF' can return overestimates, as the localisation which explicitly prevents full lookback (and keeps computational cost below $O(n^2)$) causes the 'C' in 'BCF' to match with the 'C' in 'DCD': 'ADC' is considered a substitution of 'ABC' before the algorithm can realise that 'BC' is still intact in the string, and 'DCD' is merely an insertion. As a result, the later 'B' is considered an insertion, as it no longer matches anything, and the distance is overestimated by one. Synthetic tests showed this overestimation to be minor; tests against Levenshtein on a tiny subset of Wikipedia data (a node's first few hundred revisions, thus under heavy editing) show it to be larger, with errors in the tens, and a peak error of over two-hundred. The reason for such large errors is unclear, as the resynchronisation approach should also keep *error* localised, but it does not greatly affect the result for the purpose of minor/major determination: the majority of changes were correctly classified.

Identifying changes to links, etc. was significantly simpler, and merely required comparing the sets of links identified by the parser across two revisions. These categorisations yielded simple information on which kinds of changes were made by each revision, and removed much of the 'bulk' of the dataset (the revision texts); as a result, simple scripts could then handle the data to aggregate it into various groupings in memory, so as to produce graph data and statistics for analysis. Gnuplot[4] was used to plot the graph data into graphics as part of the build process for this paper.

We identified the following non-mutually-exclusive groupings to usefully categorise edits:

Revert Edit which simply undoes a previous edit.

Content Major (nontrivial) edit of the page content.

Minor Minor (trivial) edit of the page content.

Category Edit to the categories of a page.

List of Edit to a page which is an index to other pages.

Indexing Edit to categories or listings, possibly both.

Template Edit to the templates used by a page.

Page link Edit to an internal page link.

URL link Edit to a WWW URL link; usually external.

Links Edit to page or URL links.

Link only As 'links', but excluding major edits.

Hyperstructure Any hypermedia change: indexing, linking, or template.

We expand upon the definition and significance of these groups as needed in section 6.

[4] `http://www.gnuplot.info/`

Algorithm 1 'Sliding window' string distance metric

```
procedure StringDistance(A, B)
    proc ← 0                          ▷ No. of chars. of string processed
    procstr ← NEITHER                 ▷ Last string aligned upon
    dist ← 0                          ▷ Difference accumulator
    nearA ← farA ← A                  ▷ Near and far pointers
    nearB ← farB ← B
    Let endA be the beyond-last character of buffer A,
        and endB beyond B
    procedure Scan(near, far)
        for scan ← near to before far do
            if Chars. at scan and far same then
                return scan
        return false
    repeat
        synfarA ← Scan(nearA, farA)
        synfarB ← Scan(nearB, farB)
        if synfarA ∨ synfarB then▷ Missed alignment
            if synfarA is further into A than synfarB
                is into B then
                farA ← synfarA
            else
                farB ← synfarB
        else if synfarA then
            farA ← synfarA
        else if synfarB then
            farB ← synfarB
        if Chars. at farA and farB same then
            ▷ Aligned; calc. nears after proc. point
            enA ← Min(nearA, A + proc − 1)
            enB ← Min(nearB, B + proc − 1)
            ▷ Unaligned lengths
            unA = positive dist. from enA to farA
            unB = positive dist. from enB to farB
            procedure Align(un, far, buffer, other)
                distance ← distance + un
                proc = far's distance into buffer
                if procstr = other then
                    proc ← proc + 1
                procstr ← buffer
            if unA > unB then
                Align(unA, farA, A, B)
            else
                Align(unB, farB, B, A)
            if farA = endA then                ▷ Ending
                distance ← distance+ distance between
                    farB and endB
            else if farA = endA then
                distance ← distance+ distance between
                    farA and endA
            else                    ▷ Advanced with closed window
                nearA ← farA ← farA + 1
                nearB ← farB ← farB + 1
                proc ← proc + 1
        else                        ▷ Not aligned; widen windows
            if farA ≠ endA then
                farA ← farA + 1
            if farB ≠ endB then
                farB ← farB + 1
    until farA = endA ∨ farB = endB
    return dist
```

5. TOOLS DEVELOPED

To process the sizable dataset, we created a set of small, robust, stream-based tools in C. Stream-based processing was a necessity, as manipulating the entire data in memory at once was simply infeasible; instead, the tools are intended to be combined arbitrarily using pipes. We used standard compression tools to de- and re-compress the data for storage on disk, else the verbosity of the XML format caused processing to be heavily I/O-bound.[5] The open source Libxml2[6] library was used to parse and regenerate the XML via its SAX interface. A selection of the more notable tools:

dumptitles Converts a MediaWiki XML dump (henceforth, "MWXML") into a plain, newline-separated, list of page titles. Useful for diagnostics, e.g. confirming that the random subset contains an appropriate range of pages.

discardnonart Reads in MWXML, and outputs MWXML, sans any pages which are in a namespace; pedantically, due to the poor semantics of MWXML, those with colons in the title. This implements the "trim to articles" step of figure 1.

randomsubset Reads and writes MWXML, preserving a random subset of the input pages. In order for this to be O(1) in memory consumption, this does not strictly provide a given proportion of the input; instead, the control is the probability of including a given page in the output. As a result, asking for 50% of the input *may* actually yield anywhere between none and all of the pages: it is just far more likely that the output will be around 50% of the input.[7]

categorise Reads MWXML and categorises the revisions, outputting results to a simple XML format.

cataggr A Perl script which processes the categorisation XML to produce final statistical results and graph data. By this point, the data are small enough that a SAX parser is used to build a custom in-memory document tree, such that manipulation is easier.

The tools are available under the open source MIT license, and can be retrieved from `http://users.ecs.soton.ac.uk/prb/phd/wikipedia/` to recreate the experiment.

6. RESULTS

Because of the known error margin of the approximation of Levenshtein distance, we computed results from both genuine and approximated distances on the 0.01% subset, so as to discover and illustrate the effects of approximation; the computational cost difference between the algorithms was significant: two-and-a-half hours for genuine, eight minutes

[5]Specifically, GNU Zip for intermediate; bzip2, as originally used by Wikipedia, made processing heavily CPU-bound.

[6]`http://xmlsoft.org/`

[7]A better algorithm, which is O(1) with regards to total data size, but O(n) with regards to subset size, is to store a buffer of up to n pages, and probabilistically replace them with different pages as they are encountered. However, even this would be prohibitively memory intensive on statistically significant subset sizes, as each page may have thousands of revisions, each with thousands of bytes of text, all of which must be copied into the buffer.

Edit type	Proportion
Categories	8.71%
Lists	1.97%
Overhead	10.56%

(a) 0.01% subset

Edit type	Proportion
Categories	8.75%
Lists	3.72%
Overhead	12.34%

(b) 0.04% subset

Table 1: Proportions of edits related to index management

for approximated. Results were then generated from the more statistically significant 0.04% subset (27 hours). This latter subset contained some pages on contentious topics, which had seen large numbers of revisions as a result.

6.1 Index management

Table 1 shows the proportions of edits in categories pertaining to index management. "Categories" are changes to the categories in which a page was placed. "Lists" are any change to any 'List of' page; these pages serve as manually-maintained indices to other pages. "Overhead" are changes which fall into either of these categories: because they are not mutually exclusive (lists may be categorised), it is not a sum of the other two values. Because these metrics do not consider the change in 'content' magnitude of a change, they are unaffected by the choice of string distance algorithm.

The ten percent overhead shows a strong case for the need for stronger semantics and querying on Wikipedia; this is one of the key goals, and expected benefits, of the Semantic MediaWiki project. While virtually every 'list of' node could be replaced with a query on appropriate attributes, the gain in category efficiency is harder to measure. Any semantic wiki must still be provided with categorisation metadata such that the type of pages can be used to answer such queries. However, some improvement is to be expected, as there are current Wikipedia categories which could be inferred: either because they are a union of other categories (e.g. 'Free software' and 'Operating systems' cover the existing category 'Free software operating systems') or because they are implied by a more specialised category, and no longer need to be explicitly applied to a page.

The increase in list overhead seen in the larger subset is likely a result of having a more representative proportion of 'List of' pages. Otherwise, the results are largely consistent across sample sizes.

6.2 Link management

Table 2 shows categories related to the management of links. "Links" refers to edits which changed either page-to-page or page-to-URL links. "Links only" refers to such edits *excluding* those edits which also constituted a 'major' content change: they are edits concerned only with links and other structure. "Hyperstructure" is the category of edits which changed any of the navigational capabilities of the wiki: either categories, 'List of' pages, links, or templates. "Content" is simply the category of 'major' edits.

The overestimating effect of the approximate string distance algorithm can be seen as a greater proportion of edits being considered 'major', with a knock-on effect on reducing the ratios of over edits over content edits. However, the results are consistent between the 0.01% subset with the approximated string distance, and the sample set four times the size. As a result, it would appear that the smaller size

Category	Registered	Unregistered	Total
List of	1,146	453	1,599
Revert	4,069	679	4,748
Category	6,121	954	7,075
URL link	5,548	2,977	8,525
Indexing	7,174	1,397	8,571
Template	7,992	1,330	9,322
Content	10,275	4,182	14,457
Minor	13,776	9,961	23,737
Link only	20,969	7,877	28,846
Page link	27,205	8,871	36,076
Links	29,671	10,606	40,277
Hyperstructure	38,358	11,701	50,059
Total	57,463	23,733	81,196

Table 3: Categorisation of edits for 0.01% subset, Levenshtein

of the sample set has not introduced significant error in this case, and it is reasonable to assume that a Levenshtein distance comparison of the larger dataset would yield similar results to the 0.01% subset. Therefore, further discussion will focus on the 0.01% subset with Levenshtein distance results.

These figures show the significance of hyperstructure to Wikipedia, to a surprising degree. While we expected that link editing would prove a substantial proportion of edits compared to content, we did not anticipate that *twice as many edits change links alone than those that change content.* Most link changes were page links—those to other pages on the wiki, or metawiki—as opposed to URL links to arbitrary webpages (in some cases, pages on the wiki with special arguments). 36,076 edits modified the former, but only 8,525 the latter.

With such a proportion of editing effort being expended on modifying links on Wikipedia, there is a clear need to improve this process. Introducing richer hypermedia features to wikis, such as generic links, should prove one possible improvement. Generic links are links whose endpoints are defined by matching on criteria of the document content: a basic example being matching on a particular substring. A generic link can specify that a page's title should link to that page, rather than requiring users to manually annotate it: some early wiki systems offered this capability, but only for page titles which were written in the unnatural 'CamelCase' capitalisation. Advanced examples such as local links, present in Microcosm [3, 4], can specify scope limits on the matching. This would help with ambiguous terms on Wikipedia, such as 'Interval', which should be linked to a specific meaning, such as 'Interval (music)'.

6.3 Overall editing distribution

Table 3 shows the categorisation of all edits in the 0.01% dataset, using Levenshtein for string distance, for registered and unregistered users. Note that the edit categories are not mutually exclusive, thus will not sum to the total number of edits by that class of user. "Minor" is the category of edits which did not appear to change anything substantial: either the information extracted from the markup remains the same, and the plaintext very similar; or a registered user annotated the edit as minor. Notably, over 5% of edits are reverts: edits completely rolling back the pre-

Edit type	Proportion
Links	49.60%
Links only	35.53%
Hyperstructure	61.65%
Content	17.81%
Edit type	**Ratio over content**
Links	2.79
Links only	2.00
Hyperstructure	3.46

(a) 0.01% subset, Levenshtein

Edit type	Proportion
Links	49.60%
Links only	23.36%
Hyperstructure	61.65%
Content	35.60%
Edit type	**Ratio over content**
Links	1.39
Links only	0.71
Hyperstructure	1.73

(b) 0.01% subset, Approximated

Edit type	Proportion
Links	49.56%
Links only	25.24%
Hyperstructure	61.90%
Content	35.99%
Edit type	**Ratio over content**
Links	1.38
Links only	0.70
Hyperstructure	1.72

(c) 0.04% subset, Approximated

Table 2: Proportions of edits related to link management

vious edit; this implies that a further 5% of edits are being reverted (presumably as they are deemed unsuitable).[8] A substantial amount of effort is being expended merely keeping Wikipedia 'stationary'.

Figure 2 demonstrates the distribution of users over the total number of edits they have made, in the vein of the Swartz study [10]. There is a sharp falloff of number of users as the number of edits increases (note the logarithmic scale on both axes): by far, most users only ever make very few edits, whether registered or not. Unsurprisingly, registered users tend to make more edits overall, and unregistered users are dominant at the scale of fewer than ten edits.

Figure 3 breaks the low-edit end of this distribution down by basic categories. It is interesting to note that, other than being in close proximity (e.g. "content" and "page link"), the lines do not have any definitive overlaps: the breakdown of edits is consistent regardless of the number of edits the user has made. Users who have made 70 edits have made edits in the same relative proportions (i.e., more "revert" than "list of") as those who have only made five.

Figure 4 shows how the magnitude of edits breaks down by the number of edits of that magnitude, again in the vein of Swartz [10]. Because this is clearly sensitive to the string distancing algorithm, the 0.01% subset was used, with a focus on Levenshtein: the approximate distance for all users is shown as a sparsely dotted line with a consistent overestimate. These results are largely unsurprising: registered users make larger edits, and most edits are small, with the count rapidly falling off as magnitude increases.

6.4 Limitations of detection

There are, unfortunately, several kinds of 'overhead' costs which simply cannot be detected in a computationally feasible manner by this approach. For example, MediaWiki supports a feature called template 'substitution', which actually imports the template, with parameter substitution performed (with some caveats), into the source text of the including node. It is important to note that the relationship between the including and included nodes is lost, and that the benefits of re-use (such as storage efficiency and later corrections) are not available. The information regarding the origin of the text is also lost without manual documentation effort, including any parameters required for the more complicated templates. Because use of this feature is not semantically recorded by MediaWiki, it is largely indistinguishable from the addition of a paragraph of wikitext. As a result, it is not then possible to evaluate the cost of maintaining or documenting these substitutions once the link to the original template has been lost.

It is also not computationally feasible to detect the pattern of a user performing the same fix on multiple pages, which would identify the cost of inadequate, or underused, transclusion. Transclusion is an inclusion-by-reference mechanism, where a selected (fragment of a) document is included 'live' into another, greatly facilitating re-use.

In Wikipedia, it is often desirable to accompany a link to a page with a short summary of that page's topic. In particular, Wikipedia has many cases where articles include a summary of another article, along with a "main article" link. The 'London' page[9], for example, has many sections which consist primarily of summaries of more detailed pages, such as 'Education in London'. However, without some form of transclusion or composition to share text, if the main article's summary changes—possibly because its subject changes—this change must be replicated manually out to any page which also summarises it. A transclusion mechanism would allow a single summary of the subject to be shared by all pages which reference it, including the main article on the subject, if desired.

For example, the 'Education in London' page may begin with a summary of its topic, highlighting the most notable institutions and successful research areas. The article on 'London' may then, within its 'Education' section, transclude this summary from the 'Education in London' page. Should the summary be updated, perhaps because a University gains significant notability in a new research area, this change would be automatically reflected in the 'London' page, as it is using the same text.

While MediaWiki's templates do function as transclusion, they are not employed for this role: common usage and development effort focus on their use as preprocessing macros.

7. CONCLUSIONS

The experiment consisted of the non-exclusive classification of edits made throughout the history of Wikipedia, a large and public wiki system. Classifications included both the areas of "text editing" (assumed to be primarily maintaining the *information content* of Wikipedia: its encyclopædic articles), and "link editing" (maintaining the *navigational structure* of the content). The hypothesis, that link

[8] Actual figures may vary in either direction: this does not detect rollbacks to versions earlier than the immediately preceding version, and 'edit wars' of consecutive rollbacks *will* be entirely included in the first 5%, not belonging in the latter.

[9] `http://en.wikipedia.org/w/index.php?title=London&oldid=155695080`

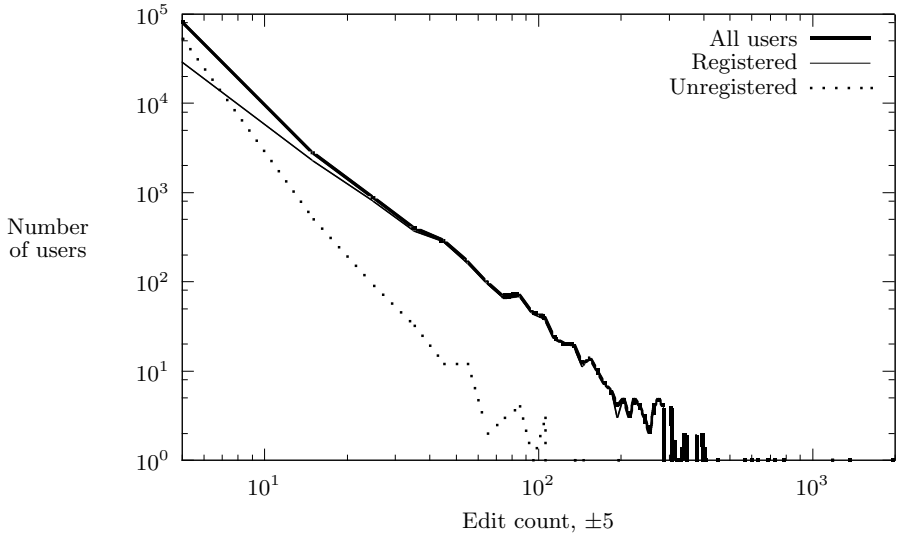


Figure 2: User distribution over total number of edits made; 0.04% subset

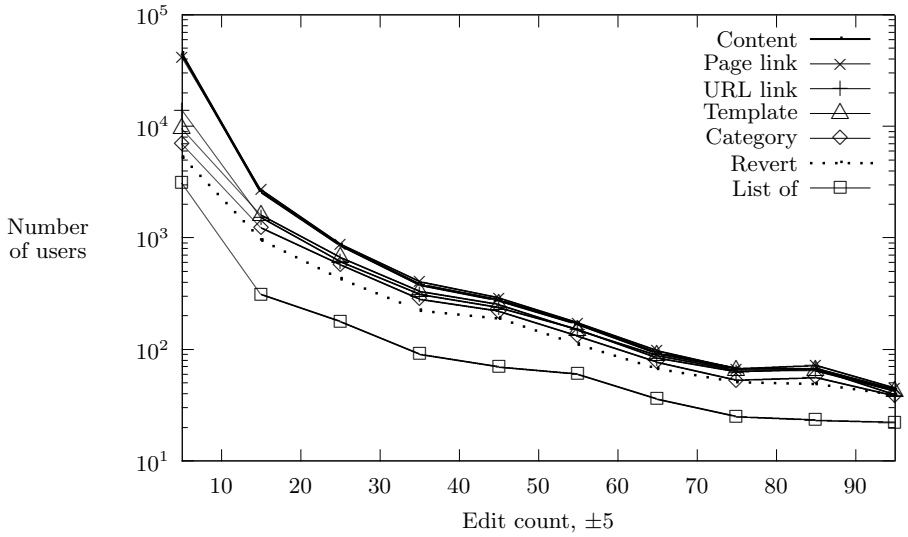


Figure 3: User distribution over total number of edits made, by category; 0.04% subset

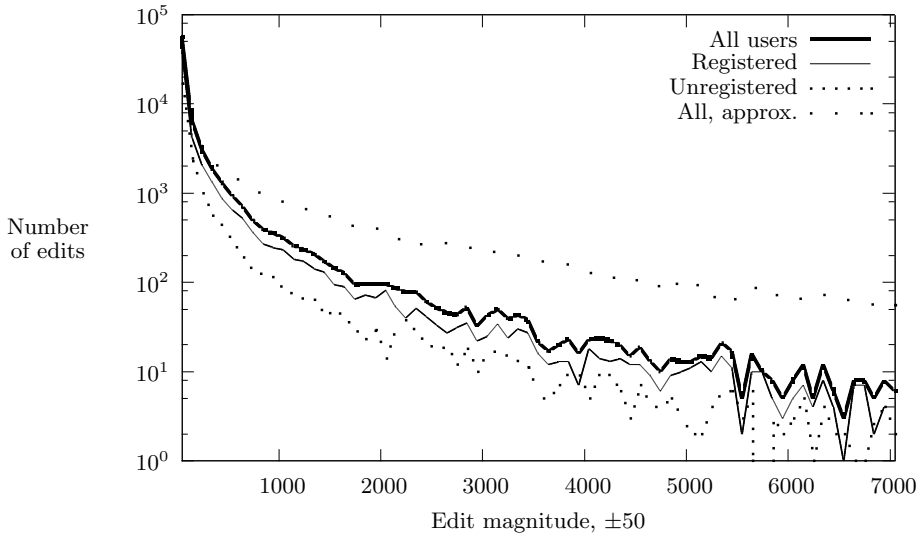


Figure 4: Edit distribution over magnitude of edit; 0.01% subset

editing formed a substantial proportion of total editing effort, which may potentially be automated, was supported by the results. Twice as many edits changed links alone, not affecting the article text. Edits which maintained manual indexes of pages constituted approximately a tenth of total edits.

We are continuing this work with a more detailed, small-scale experiment, to understand better the patterns of real-world wiki editing. It is being treated as a knowledge elicitation task, to gather information on the mental processes behind wiki editing: information on the tasks editors set themselves, and how their actions are used to achieve them. Our long-term goal is to continue this research by means of development and evaluation of a prototype system, informed by these studies, which can be used to test the hypothesis that increased hypermedia features actually result in benefits such as a decrease of editing overhead.

8. REFERENCES

[1] T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann. World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 1(2):74–82, 1992.

[2] V. Bush. As We May Think. *The Atlantic Monthly*, 176:101–108, Jul 1945.

[3] H. Davis, W. Hall, I. Heath, G. Hill, and R. Wilkins. Towards an integrated information environment with open hypermedia systems. In *ECHT '92: Proceedings of the ACM conference on Hypertext*, pages 181–190, New York, NY, USA, 1992. ACM Press.

[4] A. M. Fountain, W. Hall, I. Heath, and H. Davis. MICROCOSM: An open model for hypermedia with dynamic linking. In *European Conference on Hypertext*, pages 298–311, 1990.

[5] F. Halasz and M. Schwartz. The Dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, 1994.

[6] M. Krötzsch, D. Vrandečić, and M. Völkel. Wikipedia and the semantic web - the missing links. In *Proceedings of the WikiMania2005*, 2005. Online at `http://www.aifb.uni-karlsruhe.de/WBS/mak/pub/wikimania.pdf`.

[7] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, Feb 1966.

[8] F. Manola and E. Miller. RDF Primer. Technical report, W3C, Feb 2004.

[9] T. Nelson. *Literary Machines*. Mindful Press, Sausalito, California, 93.1 edition, 1993.

[10] A. Swartz. Who Writes Wikipedia? Online at `http://www.aaronsw.com/weblog/whowriteswikipedia`, Sep 2006.

StYLiD: Social Information Sharing with Free Creation of Structured Linked Data

Aman Shakya
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, Japan 101-8430
shakya_aman@nii.ac.jp

Hideaki Takeda
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, Japan 101-8430
takeda@nii.ac.jp

Vilas Wuwongse
Asian Institute of Technology
Klong Luang, Pathumthani,
Thailand 12120
vw@cs.ait.ac.th

ABSTRACT

Information sharing can be effective with structured data. The Semantic Web is mainly aimed at structuring information by creating widely accepted ontologies. However, users have different preferences and evolving requirements. It is not practical to attempt perfect schema definitions with strict constraints. Creating structured formats should be a collaborative and evolutionary process. Social software motivates wide participation by providing easy interface. We propose a system called StYLiD for sharing a wide variety of structured information. Users freely define their own structured concepts. The system consolidates different versions defined by different users. The attributes of the different concept versions are aligned semi-automatically into a single unified view. Popular concepts gradually emerge from the concept cloud and stabilize. Concept definitions are flexible. An attribute value can take a literal or a resource URI and the suggestive range does not constrain the contributors. StYLiD generates unique dereferenceable URIs so that data items can form a linked data web. Structured data is embedded in machine readable form using RDFa. Search and browsing features are provided to utilize the structured data and consolidated concepts.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

General Terms

Design

Keywords

Structured data, information sharing, social semantic web, concept consolidation, collaboration, linked data, RDFa

1. INTRODUCTION

Information sharing on the Web has become a basic need in communities. We want to share a wide variety of information. It would be desirable to have some system which can facilitate the modeling and sharing of such heterogeneous pieces of information. Structuring data helps handling different types of data systematically. There are many advantages of having structured data[10, 2].

- It becomes easy to define the semantics of data and make it machine understandable so that processing can be automated.
- Information sharing becomes more effective when data is structured following common conventions.
- Search and browsing becomes more effective with structured data.
- Structured information can be easily mixed. It becomes easy to integrate information from various sources.
- We can have interoperability between different systems by forming standard formats. Even multiple structure definitions for similar data may be mapped to each other.

Thus, structured data becomes open and shared for all rather than being closed in proprietary systems. With the growing significance of structured data, the Web is rapidly moving towards a *Structured Web* which can be a transitional step towards the Semantic Web and can be fully realized with current technologies[10, 2].

Efforts for the Semantic Web have been mainly being directed towards creating standard formats in the form of ontologies. However, currently there are not many ontologies to cover the wide variety of information we may want to share[19, 22]. Even if ontologies do exist, it may be difficult to search an appropriate one for our purpose. Further, understanding and using such ontologies is not an easy task for non-technical users. Like the Web, the Semantic Web should let anybody to share information about anything. There is a long tail of information domains for which different individuals have information to share[8]. There are separate well-established solutions for dealing with the head of few popular information types. However, for the long tail, availability of software is rare and developing individual solutions every time is infeasible. Moreover, a uniform solution would be desirable for interoperability and integration.

Creating new ontologies and information systems is not easy. Data modeling is a difficult task. It should be flexible to accommodate requirements and exceptions that surface in the future. Users may need different data and varying levels of details depending upon the purpose. Moreover, people

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.

have different views and should be allowed to maintain their preferences. It is not practical to impose a single standard or strict constraints.

Thus, creating ontologies or common formats should be a widely collaborative process[19]. A small team of ontology engineers cannot take into account the wide range of data and requirements of all users. However, to have large scale collaboration and to motivate general users, information sharing systems should be easy to use and understand. Ontologies can be a by-product of the usual information sharing activities in the community.

On the other hand, social software has proven to be successful in drawing huge user participation and contribution. Tagging is successful because it is very simple and anyone can contribute easily. Systems like tagging and social bookmarking do not impose any hard constraints for sharing data. However, these systems do not provide much semantic structure to information. Though some social software systems do provide structured data, they are closed systems with less interoperability and integration with other systems.

Recently, the combination of social software with Semantic Web technology towards a Social Semantic Web has been gaining significant attention[6, 1]. However, we need more tolerant mechanisms and ways to round up inconsistencies and inaccuracies that result from the informal approach of the social web[17]. We will still have a non-standard web with multiple formats. In the web, heterogeneous or overlapping conceptualizations are bound to appear[1]. However, the problem of mapping representations is not difficult, as long as the information is structured[10]. The initial step for the Semantic Web is to generate lots of data and we should facilitate easy contribution and provide incentives. Rationalization of data can be done later[8].

We propose a system called StYLiD (an acronym for Structure Your own Linked Data) which gives users the freedom to define the structure of their own data. It is easier to define one's own quick data model than to search for suitable ontology or schema and understand it. We propose to let the users input information freely without imposing any constraints, just like tagging. Computations can be done later to consolidate similar concepts, deal with inconsistencies and align multiple definitions. Concepts can gradually converge to stability by usage in the same way as folksonomies. The quality and stability of data is maintained when many eyeballs are watching and people can vote contents. This has been demonstrated well by social sites like Wikipedia[1] and Digg[2]. Furthermore, StYLiD is an open system that can link to external data and allows others to link in for building a linked data web[3].

We discuss some use case scenarios in Section 2. We describe the StYLiD platform in detail in Section 3. Section 4 gives some details about implementation. We discuss some related works in Section 5. Finally, we conclude in Section 6 and state some ongoing and future work.

2. USE CASE SCENARIO

Fig. 1 is a use case diagram which briefly shows what the user would be able to do with the system. Some details are given below.

[1]http://www.wikipedia.org/
[2]http://digg.com/

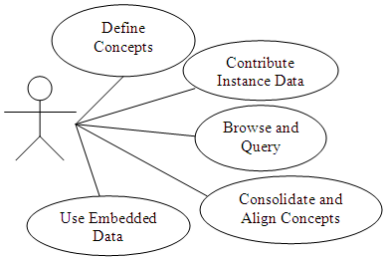


Figure 1: Use case diagram

2.1 Defining Own Structured Concepts

Suppose a user wants to share some structured information. However, he cannot find a suitable schema or any system for handling such data. He may freely register an account on StYLiD and define his own structured concept on the fly. He simply enters the concept name and a list of attributes. If a similar concept already exists in the system, he may choose to use the concept directly and enter instances or modify the existing concept to create his own version. He may modify his own concepts later and add more attributes whenever needed.

2.2 Flexible and Relaxed Data Entry

The user may easily start sharing data using his own structure definition. Any other registered user can also use his concept and contribute instance data. While entering data the system helps the user by suggesting range of values for the attributes. The user may easily pick instances from this range. However, any suitable value may be entered even though it is not in the suggested range. The user may easily type in literal values for attributes. If the user knows a resource URI for the value, it may be entered to link to that resource. The corresponding resource may also be entered later and the original entry edited to specify the URI link.

2.3 Browsing and Querying Structured Data

All the user defined concepts are visualized as a concept cloud where popular concepts are seen bigger. The user can browse different types of data with the concept cloud. When the user hovers over any concept, the attributes and description of the concept are shown so that the concept and its structure can be instantly understood (see Fig.5). This is useful to see how well defined the concept is and whether it is appropriate for him. He may wish to view only the concepts defined by him or any particular user as a concept cloud. He also maintains a personal concept collection of useful concepts, also viewed as a concept cloud. Instances of a concept can be viewed in a record view or a table view. The user may switch between these views. The user can navigate through linked data entries. The data entries may also link to external resources. The user may search data instances using a simple web-based interface by specifying the concept name and a set of attribute name, value pairs as criteria. Advanced users may directly query the system using a SPARQL query interface.

2.4 Consolidating and Aligning Concepts

Different versions of a concept defined by different users are consolidated by the system and shown as a single vir-

tual concept. The different versions are grouped together in the concept cloud. The individual concepts in a group can be identified by visible labels for the creator name and version number. By clicking on a consolidated concept, the user would be able to see all the instances of all versions. He may want to see all the instances of a concept regardless of the creator or version. He may also want to see all the instances of a concept defined by a particular user regardless of the version. He may want to see only the instances of a particular version defined by a particular user. The consolidated concept cloud offers the desired granularity.

When the concept is a single distinct concept, the table view is straightforward, each attribute displayed as a column. However, when it is a consolidated concept, the corresponding attributes of the individual constituent concepts have to be aligned first. The system automatically suggests alignments in a form-based interface. The user may update this and add mappings not suggested by the system. Then all the data can be viewed in a unified uniform table view. The user may also rename the attributes of the integrated view and hide unwanted columns if needed to get a customized view.

2.5 Utilizing Machine Readable Embedded Data

The system embeds machine understandable RDFa[3] data in the HTML posts. An RDFa aware browser would be able detect such contents and offer suitable operations for the user. Many RDFa tools and plug-ins are becoming available[4] and we may expect more powerful tools to be available in the future. The use of RDFa has also been demonstrated by recent works on semantic clipboard[15] which would allow users to copy structured data into useful desktop applications. The user may copy and paste the embedded structured data elsewhere on the Web or distribute using social media.

3. THE STYLID PLATFORM

The StYLiD platform realizes the use cases described above. It enables the users to define their own concepts on the fly and share structured data. The main contributions of the system are as follows. Details are provided in the subsections to follow.

- **Sharing structured data with user-defined concepts.** Users may define their own concepts with attributes, freely and easily, and share structured data using them. Different users are allowed to have different versions of the same concept. Users can share, reuse and refine such concept definitions.
- **Consolidation of user-defined concepts.** Multiple versions of concepts defined by different users are consolidated and corresponding attributes are aligned to produce a unified consolidated view. Popular concepts emerge out from the cloud of concepts.
- **Flexible definitions and relaxed data entry.** Users are allowed to input information freely, according to their needs and preferences, instead of attempting perfect schema definitions and imposing strict constraints.
- **Open system for creating linked data.** The system allows open access to its data using open standards. It can link both internal and external data to support a linked data web.

StYLiD is still a prototype and development is going on. A demo installation is available online[5]. Currently we are populating some sample data in the academic domain with different versions of concepts like faculty, courses, seminars, etc. Heterogeneity is common in such data because academic institutes have different systems and formats. Most of the data is being populated with the help of scrapers created using the free online service, Dapper[6]. We intend to continue using StYLiD in this domain with real users. However, the system can be installed and used for any other domain or general purpose.

3.1 Sharing Structured Data with User-Defined Concepts

The main interface of StYLiD is shown in Fig.2. The users of the system may freely define their own concepts by specifying the concept name, some description (optional) and a set of attributes. Each attribute is defined by the attribute name, description (optional) and a set of concepts as the suggested value range (optional) as shown in Fig.3. Any user may enter instance data for the concepts using the interface shown in Fig.4. An attribute of a concept can take a single value or multiple-values. Each of the values may be a literal or a resource (identified by its URI). If the value is a resource URI, a human readable label may be entered along with the URI.

Figure 2: StYLiD interface

The system allows different users to define their own concepts having the same name. Moreover, users do not need to define concepts from scratch. The user can modify an existing concept to make own version. However, users are not allowed to tamper with others' concept definitions. The system makes a copy of the concept and allows the user to make modifications on it. It keeps record of the source from which the modified concept was derived using the *dc:source* property. Users can update their own concept definitions keeping the existing instances consistent. Attributes can be

[3]http://www.w3.org/TR/xhtml-rdfa-primer/
[4]http://rdfa.info/2007/02/12/call-for-proposals-rdfa-utils-services/
[5]http://dutar.ex.nii.ac.jp/stylid/
[6]http://www.dapper.net/

Figure 3: Interface to create a new concept

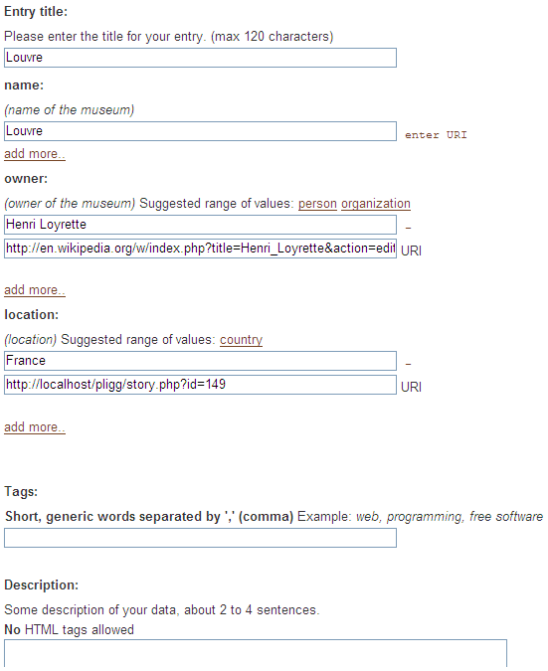


Figure 4: Interface to enter instance data

added. However, if we need to rename or delete attributes of the concept a new version of the concept should be defined to keep the existing data intact. Thus, the same user can also have different versions of his/her concept with the same name.

Structured Data Formats. The system embeds machine readable structured data in HTML using RDFa format. It also outputs the data in RDF format separately. Thus, the system produces formal machine understandable contents though the user interface is quite simple and informal like a tagging system.

A Personal Structured Data Space. The system offers every user a personal structured data space. It provides a *Concept Collection* for each user, as seen in Fig.5. Concepts created or adapted by the user are automatically added to this collection. Besides these, users can also add any other useful concepts to their collection. The users need not be overwhelmed by the huge cloud of concepts defined by the large number of users. Moreover, the concept collection is also helpful to mark the concepts that the user has been using out of numerous concepts and different versions. The concepts actually created by the user are also shown in a separate tab.

3.2 Flexible Definitions and Relaxed Entry

Creating perfect concept definitions with strict constraints is not easy and practical. It is difficult to think of all attributes and all possible value ranges at the time of concept definition. It may also be difficult to say whether an attribute value would be a literal or a resource and whether the attribute would have a single value or multiple values. While defining a concept A, if an attribute must take a resource of type concept B, we must first ensure that concept B has already been defined. If concept B has an attribute which takes resource values of type concept C, then concept C must be defined first, and so on.

Similarly, at the time of instance data entry, it may be difficult for the user to enter perfect data as mandated by a schema. All attribute values may not be known. Proper resource URIs for attribute values may not exist or the user may not be able to find it at the time. Moreover, exceptions may always exist no matter how well the schema has been designed and unpredicted new data instances may appear.

The system tries to avoid these difficulties in data modeling and data entry by allowing flexible and relaxed definitions. The concept definition may be incrementally updated later and new attributes may be added. New versions of the concept may be defined by different users or even the same user. The range of values defined for attributes, as seen in Fig.3 and 4, is only suggestive and do not impose strict constraints. Rather the system assists the user to fill data using the suggested range. The suggestive range may be updated later by including more concepts or narrowing down to refine the range. The system accepts literal values though resource values may be desirable for an attribute. Instances may be updated later to change a literal into a resource value by adding the URI. The users may input single or multiple values for any attribute as appropriate. With such relaxed data entry interface, of course, we may get some imperfect, incomplete or heterogeneous data. However, users generally enter appropriate or sensible data for their purpose. This has been evidenced by systems like tagging and wiki which accumulate large volume of good data in spite of having completely relaxed interface.

3.3 Consolidation of User Defined Concepts

Concepts defined by different users with the same name are grouped together by the system. This forms a single virtual concept which consolidates all the grouped concepts. This consolidated concept can be used to retrieve all the instances though different users have different definitions for the concept name.

If $C_1, C_2, ...C_n$ are the concepts defined by users *1, 2,... n* with concept name "C", the consolidated concept is given by $C = C_1 \cup C_2 \cup ... \cup C_n$

Further, different versions of the same concept defined by a single user are also grouped together. Thus, we can obtain all the instances of a concept defined by a user irrespective of the version.

If $C_{i1}, C_{i2}, ...C_{im}$ are the versions *1, 2,... m* of concept "C" defined by the user i, then the consolidated concept for the user is given by $C_i = C_{i1} \cup C_{i2} \cup ... \cup C_{im}$

3.3.1 Consolidated Concept Cloud

All the concepts contributed by different users are visualized together as a *Concept Cloud*, similar to a tag cloud. Better concept definitions will satisfy more users and will have more instances. Popularity of concepts is visually highlighted by increasing size. Popular concepts will receive more attention and motivate more use in turn. Thus, stable definitions will gradually emerge out from the vast cloud of concepts as more instance data are contributed. Clicking on any concept shows all instances of the concept.

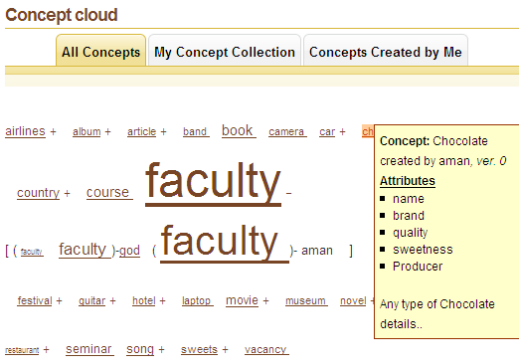


Figure 5: Concept cloud

A consolidated concept formed by grouping different versions can be expanded into a *sub-cloud*. The sub-cloud shows all the versions of the concept defined by different users, labeled with the user name. Further, in the sub-cloud, if multiple versions are defined by the same user, they are subgrouped together. In the Fig.5, the "Faculty" concept has been expanded to show two versions by the user "god" and one version by "aman". The sizes of all the different versions in the sub-cloud add up to form the size of the consolidated concept. Clicking on the consolidated concept shows all instances of all the versions of the concept. Similarly, we can also see all instances of the multiple versions of a concept defined by a single user by clicking on the user name.

3.3.2 Semi-Automatic Concept Alignment and Unification

Different concepts in a consolidated group are aligned to produce a uniform and integrated view. When the instances of a consolidated group of concepts are viewed as a table, as shown in Fig.7, the system automatically suggests alignments between the attributes of the concepts, as shown in Fig.6. Matching attributes are automatically selected in the form-based interface. Currently, the mapping is simply based on the Levenshtein edit distance[7] between the attribute labels. So slight variations on spelling and morphology are easily handled.

It is not possible to make the alignment fully automatic and accurate. Moreover, alignments may vary for different users and for different purposes. So it is desirable to have the user in loop though the system greatly simplifies the work by providing automatic suggestions. The user can complete the process by adding matching attributes that the system could not detect or modify the suggested mappings. Thus, we propose to use both machine intelligence and human intelligence for the alignment process.

A Unified View. Each set of aligned attributes can be considered as a single consolidated attribute for the consolidated concept. The system automatically fills a name for each consolidated attribute, as shown in Fig.6, though the user may rename it as desired. The user may even remove attributes from the unified view, if not required. Thus, the user can create a unified view of the consolidated concepts, customized according to his need, and view heterogeneous instance data in a uniform table.

Figure 6: Aligning the attributes of multiple concepts

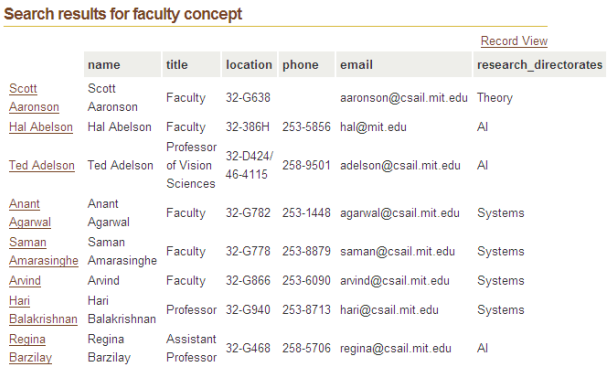


Figure 7: Table view

3.4 Open System for Creating Linked Data

The system helps in creating a linked web of data with the use of URIs. It generates unique dereferenceable URIs for each concept, attribute and instance. Each concept is uniquely identified by the concept name, its creator and the

[7] http://en.wikipedia.org/wiki/Levenshtein_distance

version number (if the same user has defined different versions of the concept).

An example URI for a concept "Car", version 2, defined by the user with ID 1 would be like

*http://www.stylid.org/stylid/concept_detail.php?concept_name=**Car_ver2_1#car***

Similarly, consolidated virtual concepts are also assigned URIs so that they can be uniquely referenced. An attribute is uniquely identified by the concept and the attribute name.

For example the URI for the price attribute of the car concept would be

*http://www.stylid.org/stylid/concept_detail.php?concept_name=**Car_ver2_1#price***

An instance is uniquely identified by the system generated ID for the instance. The URI of an instance is different from the URL of the post showing it. A concept URI dereferences to a page describing the details. An instance URI dereferences to the post showing its details. The details page contains both human readable and machine readable data.

Data instances can be linked to each other by entering resource URIs as attribute values(see Fig.4). The linked data is manifested as simple hyperlinked entries for the user (see Fig.2). However, the linking of URIs helps in the creation of a linked data web, not just hyperlinked pages. The system can link to URIs from any system on the Web. On the other hand, it allows others to link in to its data by providing unique dereferenceable URIs.

StYLiD is an open system that does not lock data into itself. Besides allowing others to link in, the system facilitates the reuse of structured data. Structured information snippets in embedded formats like RDFa may be posted elsewhere or distributed via social media. The system provides an advanced search interface, as shown in Fig.8, which can be used to retrieve instances of a concept specifying attribute, value pairs as criteria. The system also provides a SPARQL query interface for open external access.

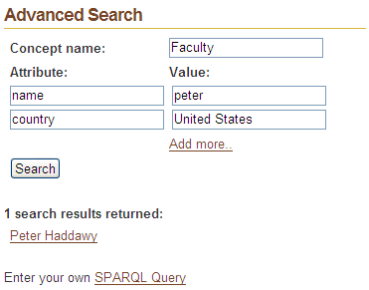


Figure 8: Advanced search interface

4. IMPLEMENTATION

Fig.9 shows the system architecture of StYLiD. It is built upon a social software platform for harnessing user contributions. The social software provides all the basic features such as content management, assessing popularity of contents, user management, social networking and communication among users. The concept management component enables the users to define their own structured concepts. The component handles the different versions of concepts defined by different users. The structured data management component gathers the instance data contributions from users. The concept management component also handles URI management by assigning each of the concepts and instances a unique dereferenceable URI. The system links structured data items using the URIs. The concept consolidation component consolidates multiple versions of a concept defined by several users. It maps the different versions by aligning attributes and provides a unified interface for the consolidated concept. The structured data embedding component embeds structured data in HTML output using RDFa. RDFa is W3C supported and a comparison with other embedded formats[8] indicates that it is a reasonable choice. The system produces snippets with embedded structured data which can be posted elsewhere. All the concepts and structured data contributed by users are stored in the collaborative data store coupled with the social software. The structured concepts and data are stored as RDF triples in a MySQL database. The system provides some services to exploit the structured data like structured browsing, search and query and allows RDFa driven features discussed in the use case scenario (Section 2.5).

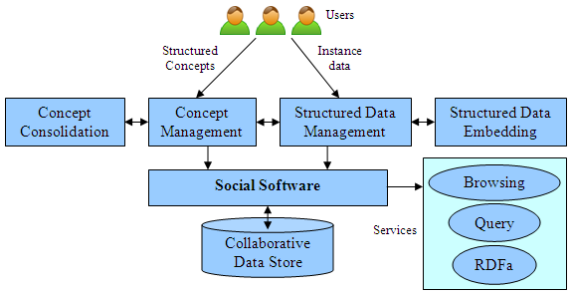


Figure 9: System architecture

StYLiD was built upon Pligg[9], a popular Web 2.0 content management system. This open source social software has a long list of useful features and strong community support and, furthermore, provides extensibility. It uses PHP and MySQL. We used the RDF API for PHP (RAP) as a Semantic Web framework to manage structured data.

5. RELATED WORK

There have been several recent works on collaborative creation and sharing of structured data on the web. Freebase[10] is one of the most prominent works. Similar to Google Base[11], it allows users to freely define their own structured types and input instance data. However, Freebase keeps the structured types defined by different users separate. It does not consolidate or relate similar concepts. Even concepts having the same name are not shown in a combined way. User defined types and domains are kept within the user's personal space and not easily promoted to the standard types and domain collection. So it is difficult to leverage the structured concepts defined by the large number of users. Moreover, it is difficult for casual users to create their own

[8]http://bnode.org/blog/2007/02/12/comparison-of-microformats-erdf-and-rdfa
[9]http://www.pligg.com/
[10]http://www.freebase.com/
[11]http://base.google.com/

types in Freebase because of the strict constraint requirements. All the attributes must have strict types and the range should be within the types already defined in the system. The attribute and range definitions cannot be altered later if some instances of the concept already exist. Further, it may also be difficult to enter instance data in Freebase because of strict schema constraints. If an attribute takes as value a resource of some type, the resource must be entered first. Although Freebase has made a lot of instance data available by scraping data from vast sources like Wikipedia and MusicBrainz, a non-existing instance must be modeled and entered by the user. Freebase interlinks instance data to each other as attribute values. However, it cannot link to external resources at the data level and it is difficult for other systems to link to Freebase data resources.

The myOntology[19] project proposes to use the infrastructure and culture of Wikis to enable collaborative and community-driven ontology building. It intends to enable general users with little expertise in ontology engineering to contribute. It is mainly targeted at building horizontal lightweight ontologies by tapping the wisdom of the community. However, myOntology is not aimed at collaboratively creating structured concepts and sharing structured data in the community based on that. Freebase and myOntology are both based on Wiki technology. Semantic Wikis, like Semantic MediaWiki[12], IkeWiki[16] and many others[12], further enhance Wikis to make the collaborative knowledge contributed by users more explicit and formal. The relations between resource pages are encoded by semantically annotating navigational links using simple syntax. However, semantic Wikis usually deal with instance data resources but do not consider forming generic schemas for structuring data. Wikis are excellent platforms for creating shared resources collaboratively. However, each concept or resource can only have a single prominent version which everyone is assumed to settle with. In practice, people may have different perceptions about the same concept. Further, users have different information sharing requirements and may need to model the same concept in different ways. StYLiD offers the flexibility and allows users to maintain their own preferences. Takeda et al.[21] had modeled heterogeneous system of ontologies by introducing *aspects*. A *combination aspect* integrates various aspects and a *category aspect* is a collection of aspects about the same thing but with different conceptualizations. They proposed muti-agent communication by translating messages across different aspects.

There had been various works on semantic blogging[4, 13, 11, 18] which exploit the easy publication paradigm of blogs and enhance blog items with semantic structure. Structured blogging[13] also embeds machine readable information in blog entries. Structured tagging techniques, like the Flickr machine tags[14], geo-tagging, triple-tags[15] or dc-tagging[16] try to inject structured information in existing social tagging platforms. However, all these systems deal with very limited types of metadata and the schemas do not evolve.

[12]http://ontoworld.org/wiki/Semantic_Wiki_State_Of_The_Art
[13]http://structuredblogging.org/
[14]http://www.flickr.com/groups/api/discuss/72157594497877875/
[15]http://geobloggers.com/archives/2006/01/11/advanced-tagging-and-tripletags/
[16]http://efoundations.typepad.com/efoundations/2006/10/dctagged.html

Works have been done on deriving ontologies from folksonomies[22, 20]. The basic ideas include grouping similar tags, forming emergent concepts from them, making the semantics more explicit, utilizing external knowledge resources and finding semantic relations. Similar techniques can also be applied on the community-grown concept cloud in StYLiD to have emergent ontologies. Folksonomies serve collaborative organization of objects. Works like MoaT (Meaning of a Tag)[17] try to make the semantics of tags explicit. However, the data objects are still left unstructured. With StYLiD users collaboratively contribute the structure too.

Revyu[7] is a reviewing and rating site where people can review and rate anything. The system generates dereferenceable URIs for *things, reviews, people* and *tags*. Data items can easily be linked with other items using URIs. Revyu produces RDF output and provides a SPARQL endpoint for query. It also exposes reviews using hReview microformat embedded in XHTML. However, most concepts are modeled simply as things. The detailed structure of the information is not modeled and different things are not differentiated.

Exhibit[8] is a lightweight framework which attempts to empower the ordinary users to publish structured information on the Web for effective browsing, visualization and mash-ups. However, authoring such pages would be cumbersome to the users. Potluck[9] is a data mash-up tool for casual users which can align, mix and clean structured data from Exhibit-powered pages. Fields can be merged by simple drag-and-drop, so that different data sources can be uniformly sorted, filtered and visualized. Merged fields are implemented as query unions. We also use a similar technique. Currently, Potluck can only handle Exhibit-powered pages and not dynamic pages and other semantic formats. The schema alignment is manual. We propose to have some automation in schema alignment instead of leaving the entire work to the users. There is a large body of research about schema matching[14] and ontology alignment[5] which can benefit us.

6. CONCLUSIONS AND FUTURE WORK

We proposed StYLiD as a single platform for sharing a wide variety of structured data. Users can freely define their own concepts. Relaxing constraints would encourage more user contribution to better meet their requirements. The task of consolidating, aligning and unifying user defined concepts can be handled by the system without bothering the users much. Although several definitions of a concept may exist, the system can provide a single consolidated view so that even heterogeneous structured data can be handled uniformly. It also facilitates the emergence of popular and stable generalized definitions. Keeping the system open and adopting URI conventions support the creation of a linked data web. Thus, even with the informal base of social software we may produce formal machine understandable structured data which can be shared, interlinked and integrated.

In the future, sophisticated schema mapping techniques[14, 5] may be incorporated to better align concept attributes automatically. On the other hand, we are working on maintaining the alignments completed by users collaboratively to utilize human intelligence too rather than relying on sophisticated computations every time. We may also allow users to save aligned unified views customized for their purpose

[17]http://www.moat-project.org/

in their own private space. Better query interfaces could be developed to query and sort instances of consolidated concepts using the combined attributes of such unified views. We may compute relations between concepts based on their structure definitions and instance data. Ideas from works on deriving ontologies from folksonomies[22, 20] may be used. Similar concepts with different names can be clustered together. Synonymous or morphological variants of concept names may be consolidated. On the other hand, ambiguous concept names may be sub-grouped by intended meaning. We can organize concepts into hierarchical domains. Scrapers may be associated to concepts for gathering abundant data from current web pages. Visual scraper creation tools may be provided so that users can easily create and share the scrapers too. We can facilitate users to contribute plugins for handling different types of structured data embedded in the pages. Other useful features, like mash-ups may be introduced to benefit from the structured data. The structured data in StYLiD may also be exposed through an API or extended RSS.

7. REFERENCES

[1] A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web. In *Proceedings of the 16th International World Wide Web Conference (WWW2007)*, Banff, Alberta, Canada, May 2007.

[2] M. K. Bergman. What is the structured web? AI3 Blog, July 2007. http://www.mkbergman.com/?p=390

[3] T. Berners-Lee. Linked data. World wide web design issues, July 2006. http://www.w3.org/DesignIssues/LinkedData.html

[4] S. Cayzer. Semantic blogging and decentralized knowledge management. *Communications of the ACM*, 47(12):48–52, December 2004.

[5] J. Euzenat, T. Le Bach, J. Barasa, et al. State of the art on ontology alignment. *Knowledge Web Deliverable D2.2.3*, 2004.

[6] T. Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Journal of Web Semantics*, 2007.

[7] T. Heath and E. Motta. Revyu.com: A reviewing and rating site for the web of data. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 895–902. Springer, 2007.

[8] D. Huynh, D. Karger, and R. Miller. Exhibit: lightweight structured data publishing. In *Proceedings of the 16th international conference on World Wide Web*, pages 737–746. ACM Press New York, NY, USA, 2007.

[9] D. F. Huynh, R. C. Miller, and D. R. Karger. Potluck: Data mash-up tool for casual users. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 2007.

[10] A. Iskold. The structured web - a primer. Read Write Web, October 2007. http://www.readwriteweb.com/archives /structured_web_primer.php

[11] D. R. Karger and D. Quan. What would it mean to blog on the semantic web? *Journal of Web Semantics*, 3(2):147–157, 2005.

[12] M. Krötzsch, D. Vrandecic, and M. Völkel. Semantic MediaWiki. In *Proceedings of the 5th International Semantic Web Conference (ISWC06)*, pages 935–942. Springer.

[13] K. Moller, U. U. Bojars, and J. G. Breslin. Using semantics to enhance the blogging experience. In *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 679–696. Springer Berlin / Heidelberg, 2006.

[14] E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal The International Journal on Very Large Data Bases*, 10(4):334–350, 2001.

[15] G. Reif, M. Morger, and H. C. Gall. Semantic clipboard - semantically enriched data exchange between desktop applications. In *Semantic Desktop and Social Semantic Collaboration Workshopat the 5th International Semantic Web Conference ISWC06*, Athens, Geogria, USA, November 2006.

[16] S. Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 388–396. IEEE Computer Society Washington, DC, USA, 2006.

[17] S. Schaffert. Semantic social software: Semantically enabled social software or socially enabled semantic web? In *Proceedings of the SEMANTICS 2006 conference*, pages 99–112, Vienna, Austria, November 2006. OCG.

[18] A. Shakya, H. Takeda, V. Wuwongse, and I. Ohmukai. Sociobiblog: A decentralized platform for sharing bibliographic information. In J. a. B. Pedro Isaìas, Miguel Baptista Nunes, editor, *Proceedings of the IADIS International Conference WWW/Internet 2007*, volume 1, pages 371–380, Vila Real, Portugal, October 2007. International Association for Development of the Information Society, IADIS Press.

[19] K. Siorpaes and M. Hepp. myOntology: The marriage of ontology engineering and collective intelligence. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, pages 127–138, 2007.

[20] L. Specia and E. Motta. Integrating folksonomies with the semantic web. In E. Franconi, M. Kifer, and W. May, editors, *Proceedings of the European Semantic Web Conference (ESWC2007)*, volume 4519 of *LNCS*, pages 624–639, Berlin Heidelberg, Germany, July 2007. Springer-Verlag.

[21] H. Takeda, K. Iino, and T. Nishida. Agent organization and communication with multiple ontologies. *International Journal of Cooperative Information Systems*, 4(4):321–337, 1995.

[22] C. Van Damme, M. Hepp, and K. Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 57–70, 2007.

m-Dvara 2.0: Mobile & Web 2.0 Services Integration for Cultural Heritage

Paolo Coppola, Raffaella Lomuscio, Stefano Mizzaro, Elena Nazzi
Department of Mathematics and Computer Science
University of Udine
Via delle Scienze 206
33100 Udine, Italy
coppola@uniud.it, raffaellalomuscio@gmail.com, mizzaro@dimi.uniud.it, elenanazzi@gmail.com

ABSTRACT

Web 2.0 marks a new philosophy where user is the main actor and content producer: users write blogs and comments, they tag, link, and upload photos, pictures, videos, and podcasts. As a step further, Mobile 2.0 adapts Web 2.0 technology to mobile users. We intend to study how Web 2.0 and Mobile 2.0 together can be applied to the cultural heritage sector. A number of cultural institutions and museums are introducing in their projects some Web 2.0 applications, but the main knowledge source remains a small group of a few experts. Our approach is different: we plan to let all the users, the crowd, to be the main contents provider. We aim to the crowdsourcing, the long tail power, as we call fuel of cultural heritage system. In this paper, we describe the m-Dvara 2.0 project, whose aim is a system that lets users to create, share, and use cultural contents including mobile context-aware features.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Science; J.5 [**Computer Applications**]: Art and Humanities; K.3.1 [**Computing Milieux**]: Computer and Education —*Computer Uses in Education*; K.4.3 [**Computing Milieux**]: Computer and Society - Organizational Impacts — *Computer-supported collaborative work*

Keywords

Culture, collaboration, cultural heritage, Mobile 2.0, museum, social, user-centered, Web 2.0, wisdom of crowd

1. INTRODUCTION

With Web 2.0 and social software we represent all web-based services with "an architecture of participation", that is, one in which users interact and generate, share, and take care of the content (`http://museumtwo.blogspot.com`). Mobile 2.0 is the evolution of mobile technology to let us "capturing the content at the point of inspiration" (`http://blog.comtaste.com/2007/06/what_is_social_in_mobile_web_2.html`), that is, in the exact moment in which the inspiration and the opportunity exist to do it. Nowadays, Cultural Heritage Organizations (museums, archaeological sites, historical towns, even libraries, etc.) are trying to understand the evolution of the Web, but they tend to stick to their traditional role, of being the sole owners of knowledge about their collections [4].

Our approach is complementary: we want to understand if a fully Web 2.0/Mobile 2.0 approach is viable for the cultural heritage sector. Indeed, in this research area, old and new conferences, e.g., Museum and the Web (`http://www.archimuse.com/conferences/mw.html`), International Cultural Heritage Informatics Meeting (`http://www.archimuse.com/index.html`), concentrate on the possible application of Web 2.0 concept and technology to museums, libraries, and other cultural heritage institutions. Web 2.0 offers a lot of useful tools:

- *Wikies* are websites that allow users to create, edit, and link web pages easily, e.g., Wikipedia (`http://en.wikipedia.org/`).
- *Blogs* are websites where entries of different types of content are commonly displayed in reverse chronological order, e.g., Blogger (`http://www.blogger.com/home`) and MoBlog:UK for mobile devices (`http://moblog.co.uk/index.php`).
- *Tagging (Folksonomy) and social bookmarking* let users to attach keywords to a digital object to describe it. Examples include del.icio.us (`http://del.icio.us/`), which launched the "social bookmarking" phenomenon or Mobilicio.us (`http://mobilicio.us/`), which is a "mashup" of Google Mobile (`http://www.google.com/mobile/`) with del.icio.us or Ma.gnolia (`http://ma.gnolia.com/`) as online bookmarking services.
- *Multimedia sharing* are services that allow storage and sharing of multimedia content, e.g., YouTube for video (`http://youtube.com/`), Odeo for podcast (`http://odeo.com`), Flickr for photo (`http://www.flickr.com/`), Twitter (`http://twitter.com/`), and Jaiku (`http://jaiku.com/`) for mobile.

By reusing and remixing these tools, static content authorities could evolve to dynamic platforms for content generation and sharing.

In this paper, we propose a set of combined Web-based services available on a unique platform, m-Dvara 2.0, that allows users to create, share, and use cultural contents. As

SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008, April 22, 2008, Beijing, China.

Web 2.0 applications gain success and become more interesting and rich with more and more users, m-Dvara 2.0 provides content on the basis of users participation and collaboration, in the very same spirit of wikipedia. The ambition of this project is to have a content repository populated by user-generated textual and multimedia content, in a new approach to improve user cultural experience through collaborating environments.

In the following sections, we first analyze several cultural heritage organizations that use Web 2.0 and Mobile 2.0 services; then, we introduce purposes and main functionalities of the ongoing m-Dvara 2.0 project, which is in the analysis stage of its development.

2. RELATED WORKS

Most museums, cultural sites, libraries, and other educational and cultural websites are not involved in Web 2.0 evolution. They are the sole provider of contents, whereas users are only consumers; for instance, Louvre Museum (`http://www.louvre.fr`), one of the first museums with a website, offers no real Web 2.0 services [2].

However, some cultural heritage organizations and some educational istitutions have introduced Web 2.0 services in their sites. In this section we provide a short summary of these projects.

- A group of US art museums are taking a folksonomic approach to their online collections: Steve (`http://www.steve.museum/`) is a collaborative research project exploring the potential for user-generated descriptions of the subjects of works of art to improve access to museum collections and encourage engagement with cultural content.

- Trant [5] has compared the Metropolitan Museum of Art in New York (`http://metmuseum.org`) terms assigned by trained cataloguers and untrained cataloguers to existing museum documentation, thus exploring the potential of social tagging: preliminary results show the potential of social tagging and folksonomies for opening museum collections to new, more personal meanings. Untrained cataloguers identified content elements not described in formal museum documentation. Tags assigned by users might help to bridge the semantic gap between the professional language of the curator and the popular language of the museum visitor [5].

- Public Library of Charlotte and Mecklenburg County (`http://plcmc.org/`) in Charlotte, North Carolina, has a teen outreach program that includes a presence in SecondLife (`http://secondlife.com`) with Teen Second Life (`http://plcmc.org/Teens/secondLife.asp`).

- Tate web site offers the youngtate section (`http://www.tate.org.uk/youngtate/`) to young people to create new learning communities, opportunities for input, and activity based on personal choice, and innovative forms of interaction with art and artists [8].

- Brooklyn Museum site (`http://www.brooklynmuseum.org/community/`) has a Community section with blogs, podcasts, forums, and a Flickr-based photos sharing service [2].

- Brooklyn College Library (`http://www.myspace.com/brooklyncollegelibrary`) uses MySpace to allow participants to post personal profiles containing their favourite books, movies, photos, and videos.

- Many projects have been developed to study how to integrate mobile devices in museum visits; [6] discusses some projects of museum covisiting with mobile device.

From these few examples is evident that Web 2.0 technologies are transforming the methods of both production of and access to cultural and educational contents, and also that the heritage sectors evolve towards user generated content. However, all these "Museum 2.0" examples also share the common approach of merely giving to the users the tools to record what the exposition had been for them, whereas a few expert members still are the main content providers. This is different from a full 2.0 approach, in which the users are given the real opportunity of creating contents in a way that makes themselves essential.

3. M-DVARA 2.0

Our approach is to let users to be not only visitors of an exposition: we want them to be the main content creators through a framework of collaboration and participation based on Web 2.0 and Mobile 2.0 technologies.

3.1 Purpose

We think users can be reliable and effective content providers, and that the wisdom of crowds is a very important source of knowledge. Can the crowd actively participate to the cultural heritage life? Can the crowd become the undisputed contents owner? We believe it is possible or at least worthwhile to try. Web 2.0 and Mobile 2.0 appropriate tools already exist and they are widespread. We propose a unique platform that uses all Web 2.0 and Mobile 2.0 technologies for our purposes: m-Dvara 2.0. m-Dvara 2.0 is an ongoing project; it is an evolution of E-Dvara, a platform storing cultural and scientific contents (`http://edvara.uniud.it/india`). The "m" and "2.0" in m-Dvara 2.0 highlight the mobile and social nature of our project. More in detail, m-Dvara 2.0 encompasses:

- a reuse of Web 2.0 technologies,
- a reuse of Mobile 2.0 technologies,
- a mix of web and mobile services,
- minimum implementation, through reuse and aggregation of Web 2.0 and Mobile 2.0 services already available online.

m-Dvara 2.0 is just an empty box with many services, whose content must be added by users, being they experts or novices. In m-Dvara 2.0 there is no central authority who publishes, owns, and controls all content.

We aim to mashup several Web 2.0 existing services (i.e., YouTube, Flickr, Blogger, etc.), in order to avoid unnecessary user efforts to interact with our system platform, and to work in an easy and comfortable way. In this way, we will provide an all-in-one familiar set of services for users. To fulfill real users requirements and expectations we will make several surveys. We plan to evaluate through several

user testings how each single service improves user experience and if it is useful. We also plan to analyse the user behavior while using the whole integrated system. Finally we are going to observe if social and Web 2.0 tools are appropriate for diffusion and perusal of cultural heritage, through evaluation of content growth and user participation level: we will observe the crowd behavior.

According to Web 2.0 concepts of remixability and aggregation, the development and adoption of standard software solutions enable websites to interact with each other by using SOAP, Javascript and any other web technology. This approach allows to interconnect websites in a more fluid user-friendly way, not only for programmers but for users as well. m-Dvara 2.0 will be based on these methodologies, examples are:

- OpenApi and OpenSocial Api (`http://en.wikipedia.org/wiki/Open_API`, `http://code.google.com/apis/opensocial/`);
- OpenID (`http://openid.net/`);
- DataPortability philosophy (`http://dataportability.org/`);

For mobile context-aware feature, we will implement a mobile service aggregator by exploiting MoBe, a framework for developing context-aware mobile applications [10]. Collaboration and participation features involve evaluation mechanisms and for this reason we propose the adoption of social evaluation. Following [7], in our system all contents can be judged by users (e.g., according to accuracy, comprehensibility, etc.). In addition, every content provider has a dynamic reliability score that depends on the scores of contents she produced. In this way, the crowd is the reviewer of its own contents.

3.2 Use Cases

System functionalities can be classified according to:

- technology being used (a user can use a mobile device, desktop, notebook, etc.),
- user location (a user can be on-site or off-site).

To introduce m-Dvara 2.0 functionalities description, we present some examples of typical use cases.

Use case 1 On-site users with a mobile device, e.g., tourists visiting a museum, an artwork exhibition, an archaeological excavation, etc.

- *Update in real-time*: the tourist can upload in real-time on m-Dvara 2.0 photo, video, audio, text about an artwork. Twitter, Jaiku technology, and/or YouTube Mobile (`http://youtube.com/mobile`) can be used to upload video.
- *Social tour*: the system can helps tourists by suggesting a tour. The tourist can request to the system an ideal tour according to her preferences, and/or tourist can select on her mobile device a tour criterion. There are three main kinds of tours: custom, dynamic and contextual tour. For custom tour we mean that system can detect user information keeping track of her actions (e.g., visited places or artworks, commented posts) or it can evaluate user's profile to set her preferences, then system process these information in order to create the user's ideal tour. A dynamic tour does not relate to user's personal information, but it depends on all users actions, thus user can decide to visit the most viewed, most commented, or most voted artworks. In other words, she can visit all the artworks that the crowd (community) advises to see. Finally, in a contextual tour, user can decide to visit only artworks about a specific topic or artworks belonging to the same artist, and so on. In addition, a tourist can change the tour criterion or she can add or remove artworks to visit from the suggested list at any time. To detect user location we intend to integrate Google Mobile with MoBe location features [10].
- *Social guides*: a cultural heritage system could be a guide. A tourist can record an artwork description as a guide and listen an audio description from her mobile device about the item she is examining. She can also access a wiki in order to read or use a screen reader to know what she needs. All different descriptions about a certain object are rated according to the crowd opinion (social evaluation). We can use, again, Twitter or Jaiku.
- *Live tagging*: the tourist can tag, using her own mobile device, the artwork she is looking at.
- *Evaluation & Rating*: the tourist can rate the artwork she is looking at. A simple rating application is automatically downloaded and executed on the tourist's mobile device, thanks to the MoBe framework [10]. The judgment is weighted accordingly to the technique proposed in [7].
- *M-Bookmark*: to bookmark from mobile devices. For this we can integrate Mobilicio.us.
- *Travel diary*: the system can keep track of artworks, monuments and places the user has seen, in order to maintain a personal travel diary.
- *M-Teach*: students can use their own mobile devices for educational lab activities.

Use case 2 Off-site users with a desktop or notebook device.

- *Wiki per topic*: the user can add contents about a topic or an object to the open wiki, e.g. Wikipedia.
- *Wiki per author*: every user can write own wiki page, e.g. Knol.
- *3D collaborative environment*: we can merge the 3D museum (e.g. Second Life) with wiki, chat, photo, and comments of users. In this way the user can visit 3D environment but she can also update wiki, talk with other visitors, write comments...
- *Blog*: the user can write a post about an artwork on her own blog, on a blog dedicated to a specific topic, or comment other blogs.
- *Bookmark*: the user can bookmark other users web-pages or artwork dedicated web-pages.

- *Personal profile and social network*: user can manage her social network, defining white and black lists. She can select her "friends" in order to create a personal sub-community. She can also suggest other users she is interested in, in order to be notified of their new posts. Similarly a user can suggest posts or themes she is interested in to be notified of their evolution.

Use case 3 Off-site users with a mobile device.

- *MoBlog*: to upload photo, video, text, audio on the blog section. We can exploit MoBlog.
- *Update in real-time*: tourist can upload in real-time photo, video, audio, text about an artefact.

To enhance user functionalities, we are considering what we call the *user events cloud*. The system will collect all available data about registered users, keeping track of all events generated (i.e., real or digital visited objects, topics of generated content, past expositions viewed, etc.), in order to create for each user an events cloud (a sort of user cultural history). We would like to use the power of the long tail of those users that know (or use) only few system functionality and help us to enjoy new features or improve already existing services (e.g., rank of content to be shown in a social tour or by social guides). All m-Dvara 2.0 functionalities will be offered to all kind of users, although we foresee a graceful degradation depending on the user context, the location, and the technology currently used.

4. DISCUSSION

In this paper we have presented how various current museum evolution projects integrate Web 2.0 services for improving user experience. We emphasized the common limitations of these "Museum 2.0" examples: they share the approach of merely providing to the users the tools to record their personal experience, while a few expert members still are the main content providers. This is different from a full 2.0 approach, in which the users participate and collaborate as the central content creators. This is the approach followed in the m-Dvara 2.0 project, whose aim is to produce a service that allows the crowd of users to control and manage the knowledge flow through collaboration and participation. We will develop an aggregator of Web 2.0 and Mobile 2.0 services for institutions of humanistic field.

Many are the problems that we are taking into account. The reuse and remixing of already existing external services involve the direct dependence from:

- their implementation - How to develop an architecture able to aggregate services featuring their own standard open interfaces and services providing personalized interfaces?
- their life - What will happen if some service does not exist anymore?

Also, copyright issues are a complex field, dependent on each nation legislation, and should be taken into account when working with cultural heritage contents. Another open question is the vandalism, that is any addition, removal, or change of content made in a deliberate attempt to compromise the integrity of the system.

Acknowledgements

The authors acknowledge the financial support of the Italian Ministry of Education, University and Research (MIUR) within the FIRB project number RBIN04M8S8.

5. REFERENCES

[1] A. Alain and M. Foggett, (2007). Towards Community Contribution: Empowering community voices on-line. In J. Trant and D. Bearman (eds). Museums and the Web 2007: Proceedings. Toronto: Archives & Museum Informatics, http://www.archimuse.com/mw2007/papers/alain/alain.html

[2] G. Crenn and G. Vidal, (2007). Les Musées Français et leurs publics a lâge du Web 2.0. Nouveaux usages du multimédia et transformations des rapports entre institutions et usagers? , in International Cultural Heritage Informatics Meeting (ICHIM07): Proceedings, J. Trant and D. Bearman (eds). Toronto: Archives & Museum Informatics, http://www.archimuse.com/ichim07/papers/crenn/crenn.html

[3] M. Middleton and J. Lee, (2007). Cultural institutions and Web 2.0. In Proceedings Fourth Seminar on Research Applications in Information and Library Studies (RAILS 4), RMIT University, Melbourne, http://eprints.qut.edu.au/archive/00010808/01/Cultural_Institutions_and_Web_2_0.pdf

[4] B. Groen, (2007). Culture 2.0, Cultuur 2.0 Online PDF Publication, http://www.virtueelplatform.nl/

[5] J. Trant, (2006). Exploring the potential for social tagging and folksonomy in art museums: proof of concept. New Review of Hypermedia & Multimedia, 12(1), 83-105, http://www.archimuse.com/papers/steve-nrhm-0605preprint.pdf

[6] Y. Laurillau, and F. Paternò, (2004). Supporting museum co-visits using mobile devices. Proceedings of Mobile HCI 2004, Glasgow, Scotland, http://giove.cnuce.cnr.it/pdawebsite/publications/MobileHCI04.pdf

[7] S. Mizzaro, (2003). Quality Control in Scholarly Publishing: A New Proposal, Journal of the American Society for Information Science and Technology, 54(11):989-1005. http://users.dimi.uniud.it/~stefano.mizzaro/research/papers/EJ-JASIST.pdf

[8] R. Cardiff, (2007). Designing a Web Site for Young People: The Challenges of Appealing to a Diverse and Fickle Audience. In J. Trant and D. Bearman (eds). Museums and the Web 2007: Proceedings. Toronto: Archives & Museum Informatics, http://www.archimuse.com/mw2007/papers/cardiff/cardiff.html

[9] P. Anderson, (2007). What is Web 2.0? Ideas, technologies and implications for education, JISC Technology and Standards Watch, http://www.jisc.ac.uk/media/documents/techwatch/tsw0701b.pdf

[10] P. Coppola, V. Della Mea, L. Di Gaspero, S. Mizzaro, I. Scagnetto, A. Selva, L. Vassena, and P. Zandegiacomo Riziò, (2005) Information Filtering and Retrieving of Context-Aware Applications Within the MoBe Framework. In Proceedings of CIR 2005 - International Workshop on Context-Based Information Retrieval, CONTEXT 2005, Paris, France.